UNIVERSIDAD LA SALLE

ESCUELA DE INGENIERIA
INCORPORADA A LA U.N.A.M.

RECONOCIMIENTO DE VOZ EN BASE A SEÑALES ORTOGONALES

TESIS PROFESIONAL

QUE PARA OBTENER EL TITULO DE:

INGENIERO MECANICO ELECTRICISTA

P R E S E N T A

JOSE LUIS PATIÑO VILCHIS

DIRECTOR DE TESIS: ING. GUILLERMO ARANDA PEREZ

1993

UNAM – Dirección General de Bibliotecas Tesis Digitales Restricciones de uso

INDICE

INDICE DE TABLAS Y FIGURAS

TABLAS

FIGURAS

INTRODUCCION

"Comuníqueme al 2-1-2-1-8-9-9 ".

Qué podría pensarse de la frase anterior, si ésta hubiera sido dicha a un aparato de marcación automática. Este reconoce un número telefónico y comunica al usuario. Para muchos tal vez es un sueño a realizar.

El reconocimiento de voz ha sido el sueño perseguido por mucha gente desde hace largo tiempo. Numerosos proyectos han sido propuestos, algunos han alcanzando una gran eficiencia; particularmente cuando se trata de un sistema monolocutor de vocabulario restringido.

Sin embargo, aún no se ha encontrado el método óptimo de reconocimiento, de otro modo, este trabajo tendría un sentido diferente.

El proyecto que se presenta en este trabajo, fue desarrollado en el Centro de Investigación y Estudios Avanzados del I.P.N., CINVESTAV. El objetivo general del proyecto es crear una operadora automática. Al hacer una llamada telefónica a CINVESTAV, la operadora automática reconoce la extensión a la que hay que enrutar; si la persona no se encuentra, entonces el mensaje se graba en el disco duro de la computadora.

La parte del proyecto que se presenta en esta Tesis sólo contempla el reconocimiento de voz que debe realizar la operadora automática. Por lo tanto, los objetivos particulares que se plantean para lograr el reconocimiento de voz, son los siguientes:

a) Basarse en el procedimiento de Gram-Schmidt para generar un espacio ortogonal.

b) Lograr el reconocimiento en un tiempo menor a un minuto.

c) Alcanzar una eficiencia mayor al 80 %.

Para comprender mejor cómo se desarrolló el proyecto de reconocimiento de voz bajo estas premisas, el trabajo de Tesis se divide en tres partes:

Parte I: ANTECEDENTES HISTORICOS DEL RECONOCIMIENTO DE VOZ.

Se recapitula cómo ha sido el desarrollo del reconocimiento de voz desde los primeros intentos al tratar de reconocer palabras aisladas hasta llegar al reconocimiento de voz continua. También se explica cuáles son las principales dificultades al reconocer, haciendo un análisis de la voz.

Parte II: REPRESENTACION DE UNA SEÑAL MEDIANTE FUNCIONES ORTOGONALES.

Comienza con una definición de las señales ortogonales; con ellas, se introduce el concepto de espacio N-vectorial aplicado a diferentes tipos de señales ortogonales para finalmente llegar a explicar el procedimiento de Gram-Schmidt.

Parte III: EL PROCEDIMIENTO DE GRAM-SCHMIDT APLICADO AL RECONOCIMIENTO DE VOZ POR COMPUTADORA.

Aquí se comienza por explicar las características técnicas que se utilizaron para implementar el método de Gram-Schmidt. Enseguida se exponen las tres partes principales del software que se utilizó para el reconocimiento: entrenamiento del sistema, construcción del espacio vectorial con el procedimiento de Gram-Schmidt y la fase de reconocimiento.

El trabajo de tesis se termina con la exposición de los resultados y las conclusiones.

I ANTECEDENTES HISTORICOS DEL RECONOCIMIENTO DE VOZ

I.1 RECONOCIMIENTO GLOBAL DE PALABRAS AISLADAS

Pensar en el reconocimiento de voz puede parecer algo inalcanzable o un tema que aún pertenece a la ciencia ficción, que sólo puede ser posible al leer un libro o al ver una película. Sin embargo, el trabajo en esta área se remonta a los años 40's y 50's. Desde esa década se ha realizado un trabajo exhaustivo que ya ha dado resultado a algunas aplicaciones comerciales.

Desde principios de siglo comenzó a ser más evidente el trabajo con señales al utilizar la transformada de Fourier. Específicamente, el trabajar con señales de voz para realizar un reconocimiento se empezó a desarrollar desde la década de los años cincuenta. Estudiar la voz se vuelve complicado ya que al tratar de reconocerla debemos tomar en cuenta :

* que la forma de señal de onda puede cambiar aunque el locutor pronuncie la misma palabra. Más aún si también cambia el locutor.

* existe ruido en el ambiente que también aparecerá en la forma de onda de la señal.

De esta manera el caso más simple es limitar el sistema de reconocimiento a un sólo locutor y con un número reducido de vocabulario.

Dentro de la historia de los reconocedores, los primeros en surgir fueron los reconocedores de palabras aisladas. Estos fueron evolucionando hasta llegar a los reconocedores de palabras continuas tal como se ve en la Tabla I.1

Los reconocedores de palabras aisladas se caracterizan por limitarse a una sóla palabra por ensayo y teniendo pausas a cada lado.

Muy pronto los estudiosos del tema se dieron cuenta que no era suficiente la información de la forma de onda y que al tratar de realizar el reconocimiento existían muchos errores.

En 1969, John Pierce, que colaboraba para la NASA en ese entonces comentaba: "He realizado pruebas sobre un reconocedor de 10 dígitos. Al realizar las pruebas con un sólo locutor encuentro una eficiencia aproximada del 90%. Sin embargo, al probar el sistema para diferentes parlantes esta eficiencia disminuye drásticamente. En mi opinión no se vislumbra una aplicación práctica para un sistema de este tipo"

En 1956 Fry y Denes publicaron que para un reconocedor, la máquina tenía que saber tanto del lenguaje como nosotros mismos. Tres dogmas aparecieron entonces para lograr el reconocimiento.

1) Lo importante es comprender el mensaje no las partes que lo componen como pueden ser fonemas o palabras.

2) La forma de onda de una señal no contiene suficiente información por sí sola.

3) Es válido utilizar cualquier fuente de reconocimiento que pueda ayudar como: reglas linguísticas, gramaticales o semánticas.

Tabla I.1

Evolución de los reconocedores de voz.

PALABRAS AISLADAS ...(Pausa)...DERECHA...(Pausa)

SECUENCIAS DE PALABRAS CONECTADAS DE
- DIGITOS, 0 ...CERO TRES CERO...
- PALABRAS EN FORMATOS ESTRICTOS
...DERECHA 30 GRADOS...

REMARCADO DE PALABRAS PRINCIPALES EN EL CONTEXTO
xxxxNIXONxxxxxxWATERGATExxxx

ENTENDIMIENTO DE LENGUAJE CONTINUO
"Cuéntame acerca de Nixon y Watergate"

LENGUAJE CONTINUO INDEPENDIENTE DEL CONTEXTO
"¿Cuál es el gran evento en Anaheim?...¿Cuántos portaaviones tiene Rusia?"

De esta manera comenzaron a surgir los reconocedores linguísticos entre otros. Con estos dispositivos podían decirse tantas palabras separadas por una pausa como la complejidad del sistema lo permitía. La complejidad de cada reconocedor iba entonces a depender del tamaño del vocabulario que se quisiera manejar, del número de locutores a los cuales se quisiera hacer accesible el sistema y de la complejidad linguística del reconocedor. Así era posible reconocer secuencias como: ....cero....uno....dos..... o bien treinta .....grados....derecha .... en donde existen restricciones linguísticas.

Si se quería reconocer palabras de manera continua, entonces el problema surgía al momento de tratar de reconocer las fronteras de cada palabra. Este problema se pudo disminuir al limitarse a reconocer secuencias de dígitos como: ....dos tres uno ..... o bien secuencias de palabras con un cierto formato como: .... veinte grados izquierda ..... Cabe notar que las secuencias de palabras aún deberían estar rodeadas de pausas. A este tipo de reconocedores se les llamó de secuencias conectadas.

Sin embargo es posible que dentro de esa secuencia únicamente se reconociera una palabra. Si esta palabra resultaba ser clave, como por ejemplo en una conversación se puede reconocer el nombre de Nixon y de Watergate, entonces se podía deducir el significado de toda la oración. Los reconocedores que utilizan esta técnica se nombran "spotting key words".

Un tipo de reconocedor más sofisticado es el de entendimiento restringido. En este modelo se hace uso de todas las consignas semánticas, gramaticales o linguísticas para realizar el reconocimiento. Así se pueden reconocer secuencias tales como "dime que pasó con Nixon y Watergate".

El modelo más complicado es el que se llama "task-independent continuous speech". La gran ventaja de este método es que se pueden reconocer cadenas de palabras aunque cambiemos la temática de una cadena a otra. Por ejemplo, "Cuál es el gran evento en México",,,,,,"Cuantos aviones tiene la armada Rusa"

La primera máquina reconocedora que realmente surgió en la historia fue un juguete llamado "Radio Rex" (Lea 1980). Este juguete estaba diseñado para saltar de acuerdo a la voz de su maestro. Se comprobó entonces que locutores con un timbre de voz parecido al del dueño o palabras semejantes a aquéllas que lo hacían saltar podían disparar su mecanismo interno.

En 1950 Dreyfus y Graf en Francia lograron armar un reconocedor que funcionaba con un tubo de rayos catódicos. Este aparato consistía de seis filtros pasobanda con diferente frecuencia central. Cada filtro estaba conectado a una bobina en la cual al variar su corriente interna provocaba la deflexión de un haz de electrones dentro de un tubo de rayos catódicos. De esta manera podía establecerse un diferente espectro en la pantalla para cada palabra pronunciada (en 1967 Yilmaz comprobó la validez de esta relación). Sin embargo este proyecto adolecía de una segunda etapa en la que no solamente se graficara, sino que pudiera realizarse el reconocimiento de los diferentes espectros que aparecían en el tubo.

Un reconocedor de gran importancia y que tuvo un gran impacto cuando apareció es el espectrógrafo que diseñaron Potter Kopp y Green en 1947. Este espectrógrafo constaba de dos ejes: un eje vertical en donde se graficaba la frecuencia de una señal y un eje horizontal en donde se graficaba el tiempo. Al penetrar la señal en el aparato, se dibujaba un espectro con diferentes intensidades de negro que daban una medida de la energía de la señal.

En 1952, Davis, Biddullph, y Balashek de los laboratorios Bell Telephone desarrollaron un reconocedor, el cual tenía por principio básico la comparación con patrones almacenados previamente. Cada una de las señales era dividida por un filtro en dos partes, por arriba y por abajo de los 900 Hz.

Posteriormente los cruces por cero eran calculados y esta densidad de intersecciones daba una medida de la energía de la señal. El espectro obtenido de esta manera era correlacionado con los patrones del sistema. Aquella señal cuya comparación tuviera una distorsión menor, era seleccionada como idéntica.

Algunos años después, en 1958, Dudley y Balashek desarrollaron un reconocedor que, a parte de utilizar las muestras patrón, poseía diez filtros que dividían a la palabra en unidades fonéticas. La eficiencia de este reconocedor creció de manera acentuada y prácticamente no tenía errores.

Alrededor de 1959 a 1960 Denes y Mathews realizaron el primer reconocedor con una computadora digital. En este trabajo ellos introdujeron el concepto de **normalización en tiempo.** Inicialmente se graba una señal patrón a una cierta velocidad de muestreo. Si la señal a reconocer tiene una mayor velocidad de muestreo, entonces automáticamente ésta es comprimida para tener una misma longitud que las de entrenamiento. Si por el contrario, la velocidad es muy lenta, entonces se busca extender esta señal. De esta manera, todas las señales tienen una longitud normalizada antes de comenzar la comparación entre señales de voz a diferentes velocidades.

A lo largo de la década de los sesenta, se desarrolló una gran gama de dispositivos específicos para el reconocimiento de voz. La idea principal era realizar un "hardware" de bajo costo y transportable que funcionara para un vocabulario pequeño. Los más importantes reconocedores de este género fueron: "Shoebox recognizer" de IBM, "Reconocedor de formato único" de Philco-Ford y otro más de RCA.

De 1958 a 1969 se buscaban mayores y más grandes aplicaciones al reconocimiento de voz: control de máquinas por voz, marcación de números telefónicos por voz, inclusive maniobar una nave espacial por voz (Kelly 1968). La capacidad de los reconocedores también fue ampliada de 50 a 500 palabras y de 1 a 10 diferentes locutores. Se reportaba una eficiencia mayor del 90%.

Medress, entre 1969 y 1972, propuso la idea de introducir el concepto de **fijativas** (Aquellos sonidos vocales de frecuencia baja), y utilizarlo en el reconocimiento: si una palabra comienza con una fijativa y enseguida hay una pausa, entonces se trata de una /s/, la pausa debe ser despreciada y enseguida debe venir una consonante sonora.

En 1972 Scope Electronics Incorporation y Threshold Technology Incorporation lanzaron comercialmente algunos reconocedores de gran éxito.

En la tabla I.2 se muestran algunos reconocedores de palabras aisladas.

En 1975 surgió un gran avance cuando Itakura introdujo la técnica de programación dinámica. Esta técnica permite realizar una normalización en tiempo de todas las señales de voz, la cual ahora es muy utilizada.

Gracias a ella, sistemas con un vocabulario de 200 palabras han sidos exitosos en un 99%. Además tiene una eficiencia del 89% sobre un sistema con 68dB de ruido en el ambiente. White y Neely posteriormente utilizaron las ideas de Itakura y lograron un 98% de eficiencia con el alfabeto hablado.

Desde 1975 comenzó a trabajarse con sistemas multilocutores que no necesitaban un entrenamiento previo. Los laboratorios Bell probaron un sistema en que los usuarios hacían marcaciones telefónicas desde sus auriculares. Se obtuvo una eficiencia del 91.6% y ésta podía aumentar considerablemente si se le permitía al usuario realizar hasta tres intentos.

TABLA I.2

REFERENCIA	AMBIENTE	VOCABULARIO	NUMERO DE LOCUTORES	NUMERO DE PRUEBAS	PORCENTAJE DE PALABRAS RECONOCIDAS
Martín Grunza, 1975	85-90 dB de ruido	34 palabras	12	9,149	98.50%
Itakura. 1975	Reconocimiento por línea telefónica	200 palabras	1	2,000	97.30%
Itakura, 1975	Reconocimiento por línea telefónica	36 palabras	1	720	88.60%
Scott, 1975	Independiente del locutor	10 dígitos	30	9,300	98.00%
Coler et al., 1977	Algoritmo "Scope"	10 dígitos	20	20,000	87,60%
Coler et al., 1977	Arbol sintáctico, dicta subvocabulario a cada punto	100 palabras	10	100,000	98.60%
Nippon Electronic Co., 1978	Dependiente del locutor	10 dígitos	4	2,400	99.80%
Nippon Electronic Co.	Dependiente del locutor	50 nombres de ciudades japonesas	1	99.8%	

En 1978 los investigadores de los laboratorios Bell reportaron un sistema con 94.4% de eficiencia que era utilizado por 30 mujeres y 25 hombres. Por otra parte, mencionaron trabajar en otro reconocedor que sería compatible con el 85% de todos los americanos sin necesidad de etapa previa de entrenamiento y con un mínimo entrenamiento para el resto.

La Tabla I.3 muestra el resumen del desarrollo histórico de la mecanización del reconocimiento de voz.

I.2 RECONOCIMIENTO DE VOZ CONTINUA

Si el reconocimiento de voz continua estuviera disponible, éste sería el método de comunicación ideal entre hombre-máquina. Ciertamente si el hombre está acostumbrado a hablar de manera fluida, entonces para cualquier usuario de una máquina sería más cómodo y más natural hablar de manera continua.

Cualquier persona está acostumbrada a decir un promedio de 150 a 300 palabras por minuto. Si se utilizan pausas, este promedio baja a menos de 125 (usualmente de 50 a 80) palabras por minuto. Esto implica que la transferencia de información del sistema disminuye drásticamente.

A finales de los años sesenta y durante la década de los setenta, fueron desarrollados varios proyectos para el reconocimiento de voz continua.

En 1966 Otten propuso la aplicación de unidades silábicas y prosódicas en diferentes estados de lenguaje (Hidden Markov Model).

Muchos proyectos se desarrollaron con la segmentación fonética de la voz (Jakai y Doshita en 1963; Hemdal y Hughes en 1967; Hughes y Al en 1969). Redd y Vicens reportaron en 1969 tener un 80% de eficiencia utilizando palabras conectadas en frases sin sentido.

Fry y Denes fueron de los primeros en utilizar restricciones linguísticas y diseñaron el "diagram frecuency" que establece una probabilidad de cuál será el siguiente fonema a reconocer y así evitar confusiones en las palabras.

Otro tipo de información linguística eran las llamadas "Distinctive features" que utilizaron: Wiren and Stubbs, 1960; Hughes, 1961; Hemdal y Hughes, 1965. Clasificaciones del tipo sonoras/insonoras, turbulento/no-turbulento, alto/bajo podían hacerse con esta técnica.

Lindgren popularizó en 1965 el método llamado "higher-level linguistic" para ser utilizado en reconocimiento. Así es que las palabras gramaticalmente correctas y con el sentido que eran esperadas podían ser utilizadas por la máquina para limitar el universo de selección de las palabras y la manera en que éstas podían ser utilizadas.

En 1971 el proyecto más ambicioso en voz que jamás haya existido se puso en marcha. La Agencia de Proyectos de Investigación Avanzada (Advanced Research Projects Agency, ARPA) del Departamento de Defensa de los Estados Unidos comenzó un proyecto de cinco años y con un costo de quince millones de dólares para desarrollar máquinas que fueran capaces de comprender frases continuas y que tuvieran un vocabulario de alrededor de 1000 palabras.

Tabla I.3

HISTORIA DE LA MECANIZACION DEL RECONOCIMIENTO DE VOZ

1947	ESPECTRO DE SONIDO
1952	DIGITOS, USANDO PALABRAS PATRONES, UN LOCUTOR
1958	DIGITOS, USANDO SECUENCIAS FONETICAS
1960	DIGITOS, NORMALIZACION EN TIEMPO
1962	"SHOEBOX RECOGNIZER" DE IBM
1964	RECONOCEDOR DE PALABRAS PARA JAPONES
1965	VOCALES Y CONSONANTES DETECTADAS EN VOZ CONTINUA * USO DE LINGUISTICA
1968	RECONOCEDOR DE 54 PALABRAS, CADENA DE DIGITOS (CODIGO POSTAL) RECONOCEDOR DE 50 A 500 PALABRAS
1969	RECONOCEDOR DE VOZ CONTINUA * CARTA DE PIERCE MENOSPRECIANDO EL TRABAJO DE RECONOCIMIENTO DE VOZ: "CIENTIFICOS LOCOS E INGENIEROS NO DIGNOS DE CONFIANZA"
1970	
1971	.....
1972	* PRIMER RECONOCEDOR DE PALABRAS COMERCIAL * 100 PALABRAS CON RESTRICCIONES FONOLOGIGAS
1973	
1974	* PROGRAMACION DINAMICA (200 PALABRAS)
1975	ALFABETO Y DIGITOS; 91 PALABRAS CON PROGRAMACION DINAMICA LOCUTORES MULTIPLES, SIN ENTRENAMIENTO
1976	SISTEMAS ARPA: HARPY, HEARSAY, HWIM;...
1977	TERMINAL DE VOZ COMPATIBLE CON CRT, REVISION DE
1978	RECONOCEDOR DE VOZ CONTINUA DE IBM

HISTORIA RECIENTE

1981	Speech Systems Inc.	Manejo del lenguaje natural.
1982	James K. Baker James M. Baker Carnegie-Melon	Desarrollo sistema: 1,000 palabras, dependiente del locutor, reconocimiento continuo, utiliza semántica/sintáctica.
1983	Audec	Marcación telefónica por voz.
1985	Audec IBM Corp. Dragon Systems	Software integrado en chips especificos para PDS. Realizan de manera conjunta driver, compilador, interfases, paqueteria de apoyo para una computadora de tipo PC.
1986	HMM Dragon Systems	Son comisionados nuevamente por ARPA para desarrollar un chip que pueda soportar las metas de la Nueva Generación de reconocimiento: 10,000 palabras, voz continua, uso de lenguaje natural, independiente del locutor.
1987	Interstate Voice Products	Crecen las aplicaciones para computadora PC. Desarrollan tarjeta para PC/XT. Tiene 240 palabras en vocabulario, ofrece 99% de eficiencia. Se tiene un paquete de entrenamiento con lista de comandos para WordStar y Lotus 1-2-3.
1988	Key Tronic	Compañía que diseña teclados, realiza reconocedor para computadora PC/XT que complementa teclado con voz, no es necesario modificar hardware.
1991	Hardware LSI	Se desarrolla hardware cada vez más poderosa y específico para LSI: en un chip se integran 280,000 transistores y se realizan 22 MOPS

Se buscaban explorar otros campos como inteligencia artificial y linguística computacional para producir una respuesta apropiada a una frase o a un discurso. Una máquina tenía que realizar las mismas tareas mentales que un humano hubiera hecho para cumplir una acción.

En el principio de los años setenta, se habían establecido avances en el área de la inteligencia artificial, así se desarrollaron programas para que una computadora juegue damas y ajedrez, para hacer deducciones lógicas e inferencias para buscar rápidamente una alternativa en un conjunto de miles. Además se desarrollaron paquetes llamados "sistemas amigables".

Surgieron teorías para caracterizar el sonido y reglas fonéticas. Se trató de establecer una estructura prosódica: entonación, tiempo, ritmo y patrones más acentuados de la voz. Todo esto podía ser acoplado con técnicas avanzadas en análisis acústico e identificación de vocales y consonantes. Un control maestro debería de coordinar todos estos procedimientos de reconocimiento.

De esta manera, ARPA comisionó un grupo de científicos para definir las metas de este ambicioso proyecto. Se contrataron cinco compañías que desarrollarían el proyecto a la mitad del tiempo programado, se realizaría una evaluación de la cual se sabría qué proyectos serían continuados hasta el fin y cuáles rechazados.

Las especificaciones del sistema eran las siguientes:

* Aceptar voz continua
* Accesible a cualquier locutor
* Tener una eficiencia mayor del 90%
* Utilizar una gramática pequeña
* Tener un diccionario mínimo de 1000 palabras

Adicionalmente el proyecto incluía investigadores que tenían como tarea el diseñar y desarrollar avanzados sistemas que pudieran facilitar la labor de reconocimiento. Se desarrollaron computadoras que podían soportar 100 millones de instrucciones internas por segundo.

Al final de 1976 este largo proyecto finalizó con la demostración de varios sistemas que podían comprender oraciones completas.

La Universidad de Carnegie-Mellon demostró dos diseños alternativos: "Harpy" y "Hearsay II"; Bolt Beranek and Newman Incorporation demostraron su sistema "Hear what I mean (HWIM)"; y Sistem Development Corporation demostró otro sistema.

De todos estos proyectos, el sistema Harpy cumplió de la mejor forma con los requisitos establecidos anteriormente. Tenía un nivel de entendimiento del 95% de las frases, reconocía cinco locutores, usaba un diccionario de 1011 palabras y una restringida gramática.

Los resultados de los cuatro proyectos anteriores se muestran en el siguiente cuadro:

TABLA I.4 Resultados del Proyecto ARPA

	HARPY	HEARSAY II	HWIM	SDC
Acepta voz continua	184 oraciones	22 oraciones	124 oraciones	54 oraciones
No. de parlantes	3 hombres 2 mujeres	1 hombre	3 hombres	1 hombre
medio ambiente	sala de computadora	sala de computadora	sala de computadora	cuarto silencioso
tipo de micrófono	estándar	estándar	estándar	buen micrófono
No. de palabras en el vocabulario	1011	1011	1097	1000
rendimiento con un error semántico inferior al 10 %	95%	91%	44%	24%

Aunque el sistema Harpy no podía ser comercializado inmediatamente, dado que aún se encontraba restringido a cinco locutores, contribuyó con muchas aportaciones que no se tenían anteriormente sobre las propiedades de los sonidos y de la voz, así como diversos adelantos computacionales.

Para 1977 investigadores de IBM pensaron en dividir la señal de voz en "Transemes" que abarcaban las muestras del centro de un fonema al centro de otro fonema. De esta manera, la voz se hacía menos variante que con unidades fonéticas normales.

Doddington de Texas Instruments desarrolló un sistema muy existoso de seguridad basado en el reconocimiento de voz. Este sistema reconocía una cadena de seis dígitos para determinar el número de identificación de una persona, enseguida se verificaban las características de la voz para corroborar los datos anteriores.

En 1978 Rabiner y Sambur de los laboratorios Bell crearon un reconocedor que detectaba los números entre dígitos continuos. La técnica consisitía en detectar partes no sonoras de la señal de voz, es decir, aquéllas en las que las cuerdas vocales no vibran, así como descensos en la energía. Estas dos condiciones indicaban una consonante.

Por esta época se desarrollaron los "Word spotter" que tenían que distinguir palabras sin ser sensitivos al cambio de locutor o distorsiones de canal en las comunicaciones. Prácticamente este método consistía en correlacionar fonemas con otros que se tenían almacenados como patrones.

En el lapso de los siguientes tres años (1988, 1989, 1990) surgen más aplicaciones específicas en el procesamiento de voz. En la tabla I.5 se muestran algunas de estas aplicaciones.

Tabla I.5

Aplicaciones recientes del reconocimiento de voz

COMPAÑIA	APLICACIONES
RCA	Control de cámara por voz: Pan, Zoom, Tilt. Posición de cámara con Joystick.Combina reconocimiento de voz (templetas), inteligencia artificial (encuentra sentido de palabras) y Robótica.
Scott Instruments Corporation	Se reconocen porciones de una palabra (alofonos). Su sonido cambia dependiendo de la posición en la palabra en la que se use. Los alófonos reflejan entonación y énfasis. Se combina Reconocimiento, compresión de voz y reproducción de voz. Se interfasa para paquetes CAP/CAM.
Speech Systems Inc.	Posee un diccionario maestro de 20,000 palabras.
Technical Instruments Corporation	Diagrama de Software para inspeccionar defectos sobre un objeto ordenado por voz.
Texas Instruments	Paquetes de desarrollo que incluyen reconocimiento de voz, síntesis y análisis, conversión texto-voz. Reconocimiento de palabras aisladas o contínuas.
Voice Industries	Generan Software para hacer desarrollos de voz en computadoras más grandes como DE VAX.
YOTAN	Integra las soluciones para un sistema de reconocimiento de voz vía telefónica. Chips LSI realizan tareas específicas de correlación de voz, reconocimiento contínuo. Inmunidad al ruido hasta de 100 dB.

COMPAÑIA	APLICACIONES
Vynet Corporation	Sistemas de grabación digital desde un micrófono, cassette o teléfono, hasta correo de voz que permite llamar o ser llamado por una computadora, dar o recibir información con voz digitalizada. Si se requiere, es posible editar frases y prompts de voz.
Westinghouse Voice Systems	Realiza sistemas de comunicación para workstation que permiten un canal de comunicación vía voz, teclado o "bar code" y poseen promting visual o por audio

I.3 QUE ES LA VOZ Y PORQUE ES DIFICIL RECONOCER

El diseño y construcción de sistemas para la comunicación por voz hombre-máquina ha sido una gran empresa que se ha perseguido durante mucho tiempo.

Se han elaborado reconocedores que pueden funcionar con diversos locutores y un número de varios miles de palabras. Inclusive comercialmente existen sistemas de reconocimiento que han tenido un éxito considerable. Sin embargo, aún no se puede afirmar que se haya logrado el nivel óptimo en el diseño de reconocedores y que éstos puedan fácilmente llevarse a cualquier aplicación.

Antes de mencionar los métodos de reconocimiento, es conveniente detenerse a observar con qué se está trabajando: Qué es la voz y porqué es difícil de reconocer. Cuando la voz se escucha, no se detectan únicamente sonidos, las señales de voz conllevan más información que sólamente la acústica.

Generalmente con el mensaje que se obtiene de la señal de voz, también existe información cultural del locutor, como puede ser de qué región proviene, o de qué clase social es. Existen además los rasgos particulares de la persona y que lo distinguen de los demás como son: su entonación, su articulación, etc.

Se puede decir que la información que se extrae de la voz es:

1) Lingufstica
2) Sociolingufstica
3) Personal

Esto debido a que cada persona puede hablar de manera diferente y posee rasgos fisiológicos únicos.

Al momento de transmitir una señal de voz, es posible dividirla en segmentos. Los segmentos acústicos más pequeños se llaman **unidades fonéticas**. Estas a su vez pueden clasificarse en vocales o consonantes fonéticas. Esta división no es la que normalmente se acostumbra manejar ortográficamente; es decir, si se piensa que las vocales fonéticas son solamente a, e, i, o, u, y las consonantes fonéticas el resto del alfabeto, se está en un error. La división anterior se hace en base a los diferentes tipos de articulación, de los cuales se hablará ahora.

I.3.1 ANATOMIA DE LA VOZ

Cada vez que una persona habla, lo logra haciendo expulsar aire desde sus pulmones. El aire de los pulmones pasa a través de las cuerdas vocales, que son dos pequeños pliegues musculares localizados en la laringe, que está situada justo detrás de la manzana de Adán. El espacio entre las cuerdas vocales es conocido como glotis. Si las cuerdas vocales están separadas (como cuando respiramos normalmente) el aire de los pulmones va a tener un paso relativamente libre hacia la faringe y luego hacia la boca. Pero si las cuerdas vocales están muy cerradas de tal manera que exista un estrecho paso entre ellas, el flujo de aire será comprimido. Tan pronto las cuerdas vocales se cierren completamente, se formará una mayor presión abajo de ellas debido a que el aire no puede pasar más. Esta presión cesará cuando las cuerdas vocales se vuelvan a abrir, el aire pasará comprimido de nueva cuenta y las cuerdas vocales permanecerán vibrando.

Los sonidos producidos con las cuerdas vocales vibrando se llaman sonoros, mientras que aquéllos en los cuales las cuerdas vocales están totalmente separadas, se llaman insonoros.

Después de que el aire pasa arriba de las cuerdas vocales, llega a lo que comúnmente se conoce como el tracto vocal. En esta parte el aire encuentra obstrucciones formadas por la lengua. Las consonantes pueden ser clasificadas de acuerdo a la manera en que esta obstrucción puede aparecer.

Las principales articulaciones realizadas en el tracto vocal se muestran en la figura I.1 y éstas son las siguientes:

1) Bilabial formada entre los labios.
2) Dental formada con la punta de la lengua y la parte superior de los dientes frontales.
3) Alveolar se hace juntando la punta de la lengua y la carnosidad arriba de los dientes.
4) Retroflex se lleva la punta de la lengua un poco más atrás de la carnosidad de los dientes superiores.
5) Palato-alveolar en la misma posición que se llevó la punta de la lengua en la articulación anterior, ahora hay que llevar la "paleta" de la lengua.
6) Palatal se realiza juntando el frente de la lengua y el paladar.
7) Velar se junta la parte trasera de la lengua y el final del paladar (llamado paladar suave).

En estas posiciones es posible realizar hasta seis diferentes tipos de articulaciones como son: paros, golpeos, laterales, fricativas y africativas.

Paro

La articulación de paro implica cerrar las articulaciones de tal manera que el aire no pueda salir de la boca. Este tipo de articulación puede definirse como nasal u oral. Si el paladar suave (velum) está abierto de tal manera que el aire pueda circular por la cavidad nasal, entonces se dice que es un paro nasal. Sonidos de este tipo ocurren, por ejemplo, al principio de las palabras *mío* y *noche*.

Si además de cerrar la boca, también se cierra el paladar suave de tal forma que no se permita salir el aire en absoluto, se habla de un paro oral. En este tipo de articulación el aire irá acumulando presión hasta que se abra la boca y ocurra una pequeña explosión. Este tipo de sonidos ocurre al principio de palabras tales como: *pie*, *gato*, *tetera*, etc.

Golpeo

Este sonido es producido si una articulación golpea a otra momentáneamente como la consonante que se encuentra a la mitad de la palabra *Betty*.

Laterales

Ocurren cuando la lengua forma una obstrucción por la parte central de la boca, dejando pasar el aire por las regiones laterales. Los sonidos al principio y al final de la palabra *lineal*, *liberal* son laterales.

Africativas

El sonido al inicio de la palabra *chocolate* es una combinación de un paro y una fricativa. Este tipo de combinaciones se conocen como africativas.

Los sonidos de las vocales, al igual que los sonidos de las consonantes pueden ser descritos en términos de las articulaciones. Específicamente éstos dependen de la forma del tracto vocal.

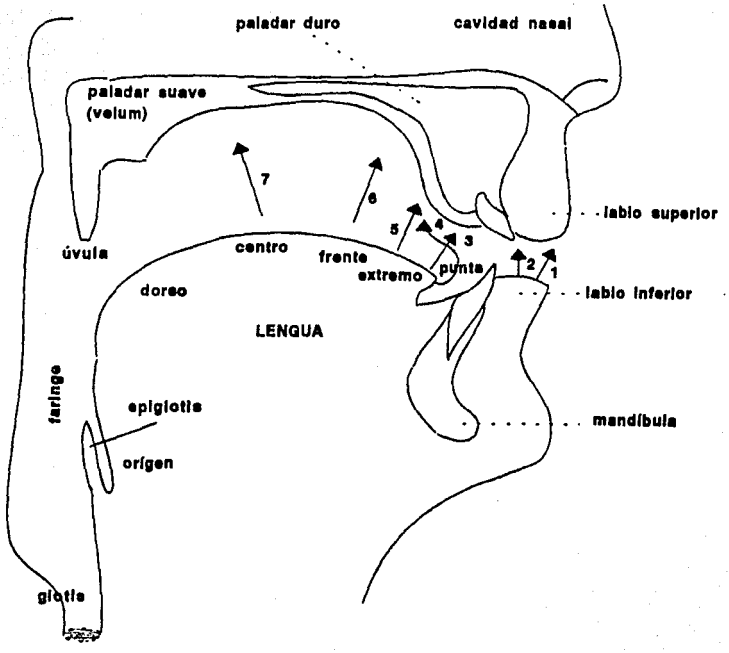


Figura I.1 Los organos vocales y el modo en el que son utilizados en algunos lugares de articulacion.

1)bilabial 2)dental 3)alveolar
4)retroflex 5)palato-alveolar
6)palatal 7)velar

Existen tres parámetros que puede decirse caracterizan completamente la forma del tracto vocal:

1) El tamaño (en cm^2) de la mínima sección transversal, A_{min}.
2) La localización de A_{min} en términos de la distancia, L (en cm), desde la glotis.
3) La magnitud de abertura de los labios, A_{lip} (en cm).

La figura I.2 muestra estos tres parámetros para dos vocales: [i] como en *gis* y [a] como en *padre*. Se puede observar que la vocal [i] tiene una área mínima significativamente lejos de la glotis, y que la vocal [a] tiene un área similar pero mucho más cercana a la glotis.

I.3.2 ACUSTICA DE LAS VOCALES

Prácticamente todas las vocales son sonoras; es decir, que éstas son producidas haciendo vibrar las cuerdas vocales. Cada vez que las cuerdas vocales se abren y se cierran, se forma un pulso de aire desde los pulmones.

Estos pulsos chocan contra el aire que se encuentra al interior del tracto vocal que entonces se pone a vibrar de acuerdo al tamaño y la forma del tracto. En el sonido de una vocal, el aire al interior del tracto vibra en tres o cuatro frecuencias simultáneamente. Estas frecuencias son de resonancia para la forma particular del tracto vocal. La frecuencia fundamental está dada por la rapidez de vibración de las cuerdas vocales; y las tres o cuatro frecuencias de resonancia permanecerán mientras no cambie la posición de los órganos vocales.

Las resonancias del tracto vocal son conocidas como **formantes**. Es posible determinar las frecuencias de los formantes, como en la figura I.3, en términos de:

(1) el área mínima transversal al A_{min}
(2) la distancia L de la glotis
(3) la abertura de los labios

La figura I.3 muestra las frecuencias de los cinco primeros formantes en función de la distancia a la glotis y suponiendo un área mínima transversal constante de 0.65 cm^2.

En la figura es posible observar que a medida que la constricción mínima se mueve hacia la glotis, algunos de los formantes decaen en frecuencia.

Esta disminución en la frecuencia es debida al incremento en la longitud del tracto vocal a partir de la constricción. Esta parte del tracto vocal actúa como un resonador produciendo un sonido con una longitud de onda que depende del largo de la cavidad.

El efecto de variar la posición de los labios, también está ilustrado en la figura I.3. Al redondear más los labios (disminuir la apertura de la boca), se traduce en un decaimiento sustancial de F2.

Para los otros formantes casi no hay variación, sólo para F3 en la región alveolar y para F1 en la faringe. Un promedio de estos formantes es para F1 = 280 Hz., para F2 = 2250 Hz y para F3 = 2890 Hz.

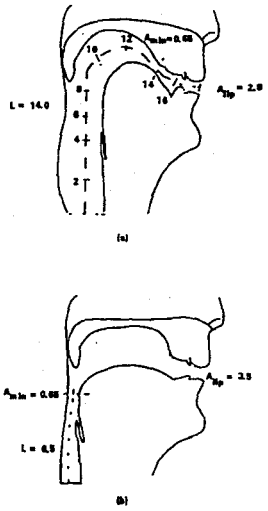


Figura I.2 Especificación de la forma del tracto vocal en:

(a) una vocal como en "gis"
(b) una vocal como en "padre"

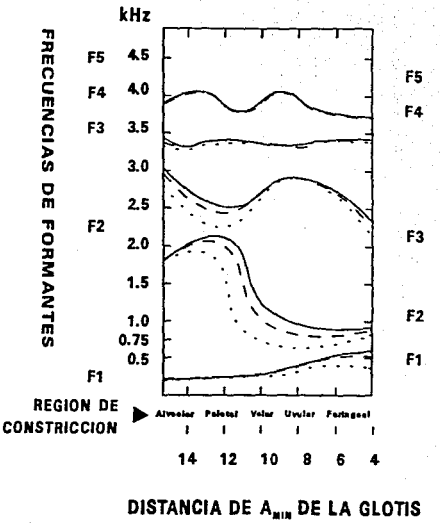


Figura I.3 Frecuencias de los cinco primeros formantes en función de la distancia a la glotis

I.3.3 PORQUE ES DIFICIL DE RECONOCER

La necesidad para reconocer segmentos de voz ha generado reglas para clasificar y codificar la voz, de tal manera que se puede almacenar, recuperar y reconocer.

En este sentido es posible clasificar la voz de dos maneras: como una transcripción fonética y de una manera prosódica.

TRANSCRIPCION FONETICA

Un fonema es una unidad abstracta que representa un conjunto de sonidos diferentes. Así, por ejemplo, los sonidos iniciales en *kiosco* y *caso* pertenecerían al mismo fonema /k/, por lo que no es este sonido el que distingue a las dos palabras. En el caso de *ver* y *ser*, "v" y "s" pertenecen a diferentes fonemas puesto que sus sonidos distinguen un par de palabras.

Se puede afirmar que un fonema es un grupo de sonidos que:

a) Deben ser iguales para el locutor.
b) Un sólo fonema no puede ser utilizado para distinguir dos palabras.
c) Un fonema es un término dado a un conjunto de sonidos.

Existe una diferencia entre lo que son vocales y consonantes en fonética. Una vocal forma el núcleo de cada sílaba y una o más consonantes pueden aparecer a los lados de una vocal. Es necesario remarcar que las vocales fonéticas son algunas más que a, e, i, o, u.

Los fonemas pueden ser clasificados en grupos que reflejen ciertas similitudes. La manera de agrupar los fonemas depende de los tipos de articulación como se vió en la sección anterior.

La tabla I.6 dá un resumen de los tipos de fonemas de acuerdo a su articulación.

Tabla I.6 Resumen de los tipos de fonemas

TIPO	FONEMAS	ARTICULACION DISTINTIVA
Vocales	uh, a, e, i, o, u, aa, ee, er,uu, ar	Vibración de las cuerdas vocales
Paro	p,t,k,b,d,g	Cavidad nasal y tracto vocal cerrados
Nasal	m, n, ng	Cavidad nasal abierta, tracto vocal cerrado
Fricativo	s, sh, f, z, v	Turbulencia del aire en el tracto vocal; éste parcialmente cerrado
Africativo	ch, j	Combinación de paro y fricativo

Como se puede observar en la sección anterior, tanto vocales como consonantes pueden clasificarse en ciertos parámetros en función de la forma del tracto vocal.

Sería factible imaginar que una vez que se tienen clasificados los fonemas y en el entendido de que son una representación no ambigua de la señal de voz, podrían ser utilizados para el reconocimiento.

Entonces basta saber, para una señal de voz desconocida, con qué fonemas se puede representar para poder comprender el mensaje.

Sin embargo, existe poca variación en ciertos fonemas y algunos otros prácticamente no contribuyen con información alguna para su reconocimiento. Para observar esto, se puede hacer un estudio de los parámetros acústicos de algunas palabras.

Los parámetros acústicos de los sonidos de voz pueden ser determinados por un análisis espectrográfico. El espectrógrafo de sonido es un intrumento que produce registros de las componentes de frecuencia como función del tiempo.

En los proximos espectrogramas que se presentarán, la escala del tiempo se encuentra en el eje horizontal, mientras que la escala vertical muestra la frecuencia en Hz. La relativa intensidad de cada componente de frecuencia es mostrada por la intensidad de las marcas negras.

La estructura acústica de las consonantes es usualmente más complicada que aquélla de las vocales. En muchos casos una consonante se especifica como un caso particular de comenzar o terminar una vocal. De esta manera, realmente no hay diferencia significativa en las consonantes {b, d, g}. Esto se puede apreciar en el espectrograma de la figura I.4 y en las formas de onda de la figura I.5

De acuerdo a los niveles de intensidad de energía podría pensarse que es la misma palabra, no existe una real distinción.

La figura I.6 muestra los espectrogramas para las palabras tan, can.

Es posible observar cómo los inicios de estas palabras que son fricativos, prácticamente no se muestran en los espectrogramas. Este tipo de consonantes complican el reconocimiento al ser difícilmente identificables.

PROSODIA

La clasificación fonética introducida anteriormente divide la voz en segmentos y la separa en fonemas. Más allá de estas divisiones, existen atributos prosódicos que se deben a la señal de voz.

La prosodia está definida como la ciencia que enfatiza en los aspectos de stress y ritmo que, por ejemplo, son intrínsecos a los versos clásicos. Estas características prosódicas también son llamadas suprasegmentales ya que están por encima de los fonemas o los segmentos silábicos.

Las características prosódicas pueden ser divididas en dos categorías básicas:

1) calidad de la voz
2) características dinámicas de la voz

En el primer grupo se toman en cuenta las variaciones que pueden existir en la voz por diferencias anatómicas.

Ya anteriormente se había mencionado la gran variedad de sonidos que se puede tener para una misma palabra con diferentes locutores. Esta gran variedad es debida en parte al tamaño de la cabeza. Un locutor con una cavidad vocal muy grande, produce vocales con una frecuencia fundamental mucho menor que aquellas personas que tienen cabeza pequeña. Las mujeres tienen frecuencias de formantes que son en promedio 17% más altas que aquéllas de los hombres. Algunas vocales son afectadas de manera significativa por el tamaño de la faringe.

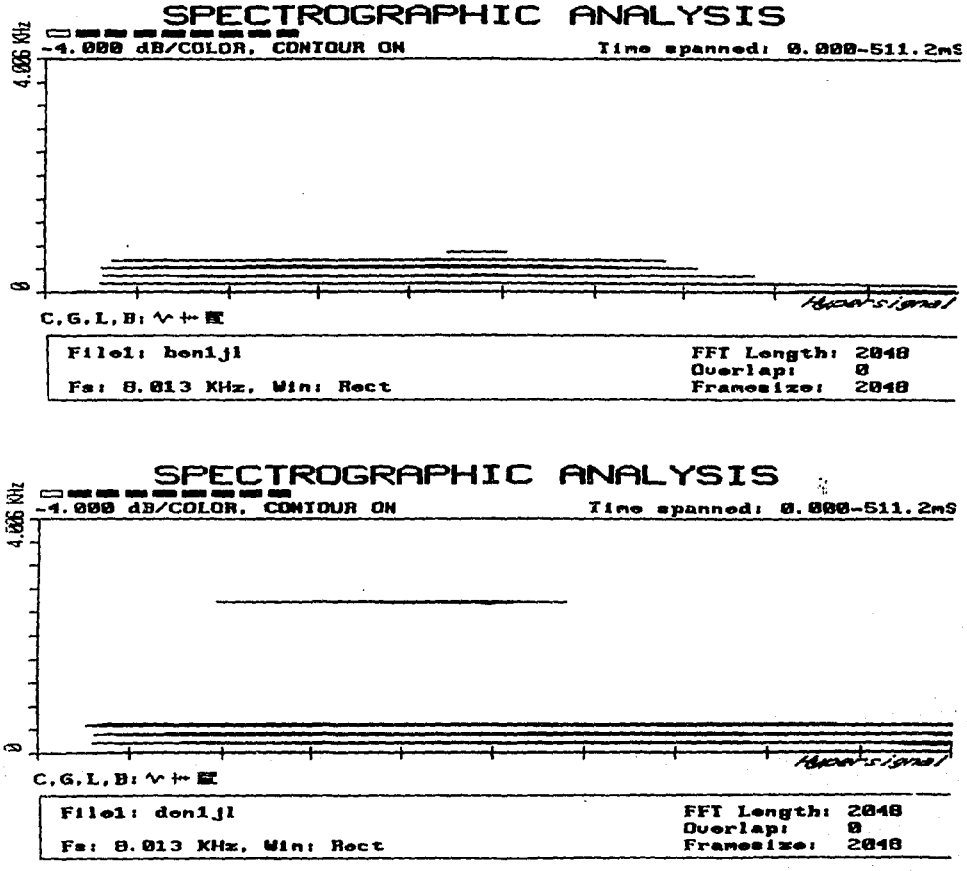


Figura I.4 Espectogramas de las palabras "ben", "den"

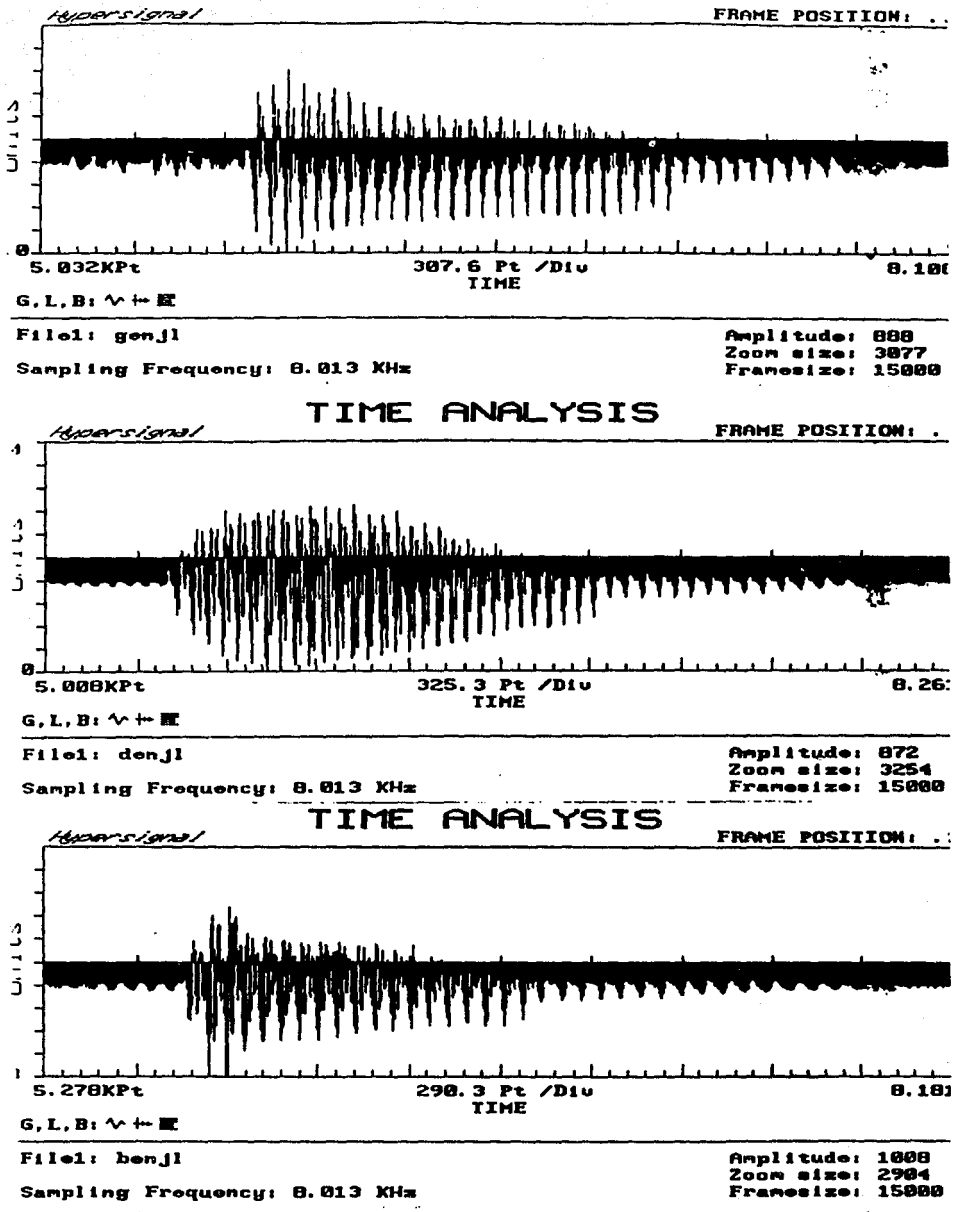
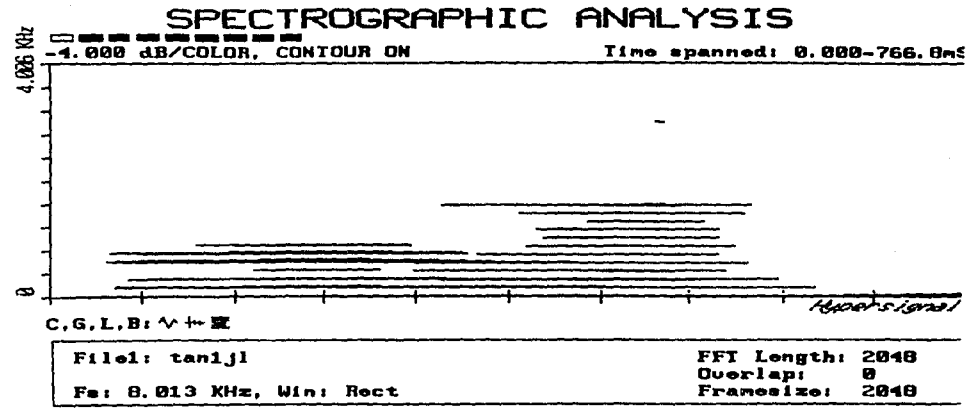


Figura I.5 Formas de onda de las palabras "ben", "den", "gen"



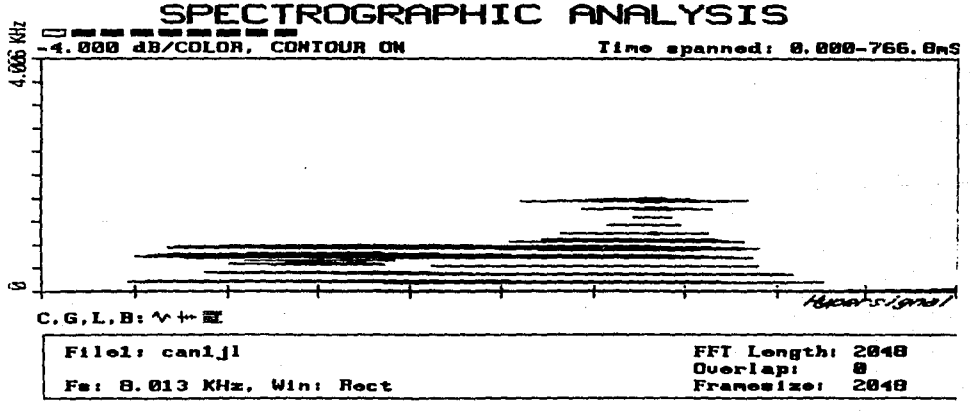


Figura I.6 Espectogramas de las palabras "tan", "can"

Dos vocales pueden tener la misma articulación, pero diferentes formantes de frecuencia. De la misma manera pueden tener diferente articulación, aunque mismos formantes de frecuencia, o en el último de los casos, variaciones tanto en articulación como en los formantes de frecuencia, siendo este último el de mayor dificultad para lograr el reconocimiento.

En la figura I.7 se muestran los espectogramas y las formas de onda de señal para la palabra *sabio*. El primer registro es de un locutor hombre hablando normalmente, el segundo registro es de un locutor mujer hablando normalmente.

Obsérvese como las tres señales parecen ser totalmente diferentes aún cuando se trate de la misma palabra. El cambio radical se debe a la variación en los parámetros que caracterizan el tracto vocal.

En el segundo grupo, las variaciones pueden ocurrir en tres dimensiones: pitch o frecuencia fundamental de la voz, tiempo y amplitud.

Las variaciones en la frecuencia de la voz producen cambios de entonación; de esta manera, si por ejemplo si al hablar con alguien se quiere remarcar la importancia de algo, se cambia la entonación. En la frase: "Compro un carro rojo", si cambiamos la entonación en *rojo*, estamos subrayando su importancia.

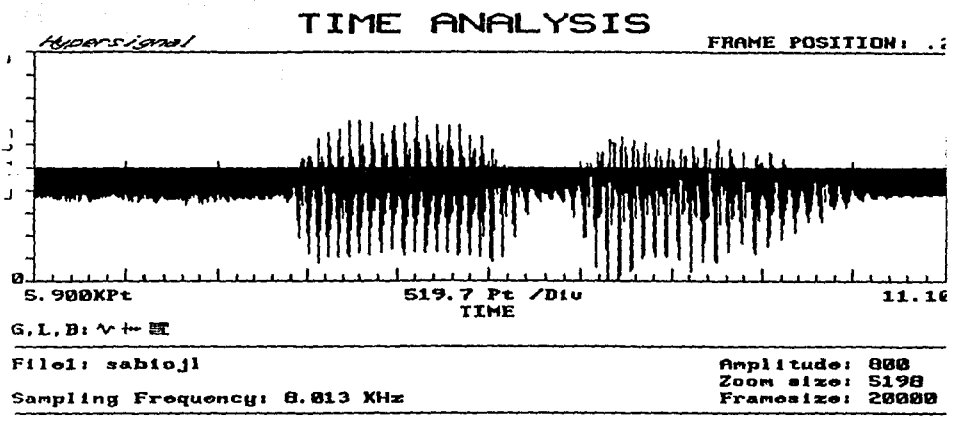
La dimensión del tiempo indica las pausas que pueden ser utilizadas por un locutor y que pueden ser significativas para el mensaje, por lo tanto, no es lo mismo decir: A...su...lado que azulado. Esta dimensión también implica el ritmo al hablar: despacio o rápido.

La tercera dimensión de amplitud es de relativa menor importancia. Las primeras dos dimensiones pueden combinarse para formar el efecto de lo que se llama "stress". Las varaiaciones que se pueden obtener juntando estas variaciones son enormes y generalmente son ocasionadas por aspectos emocionales.

Entre la gama de sentimientos audibles está: el sarcasmo, la excitación, rudeza, desacuerdo, tristeza, miedo, amor. Todos estos matices dan características muy especiales a la voz que la hacen muy difícil de reconocer.

La figura I.8 muestra las formas de onda de señal de las palabras "buenos días", variando las tres dimensiones.

1. Normal
2. Buenos....días, remarcar días
3. Buenos días, susurrando
4. Buenosdías, sin pausas prácticamente cambio en el ritmo.



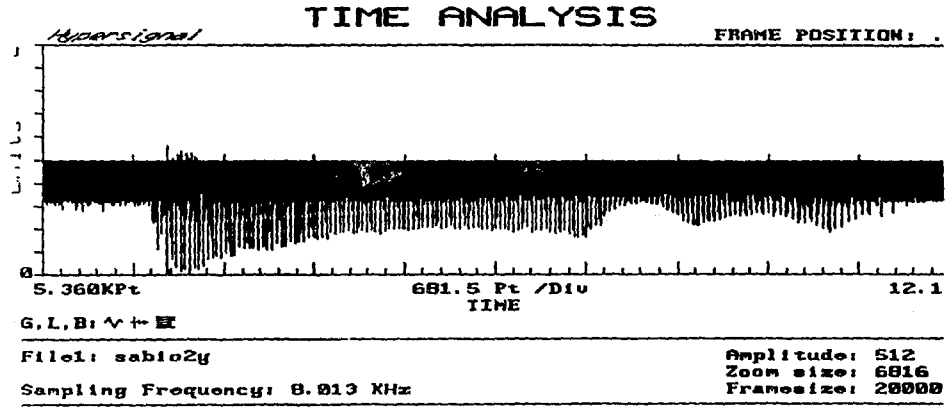
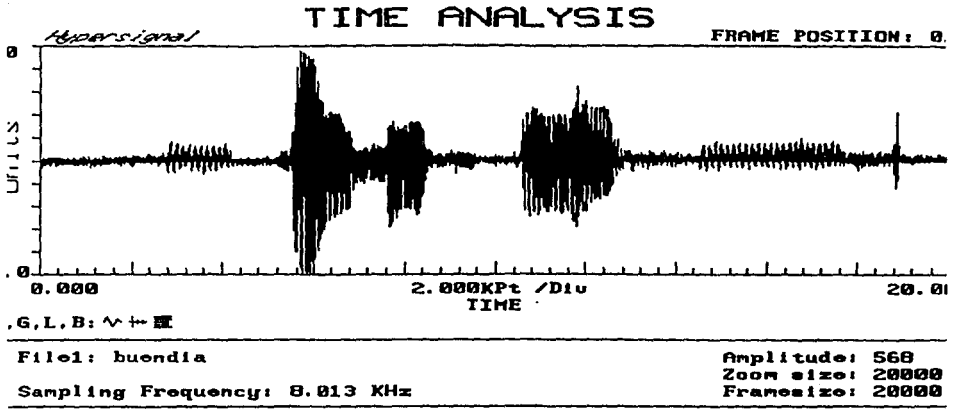
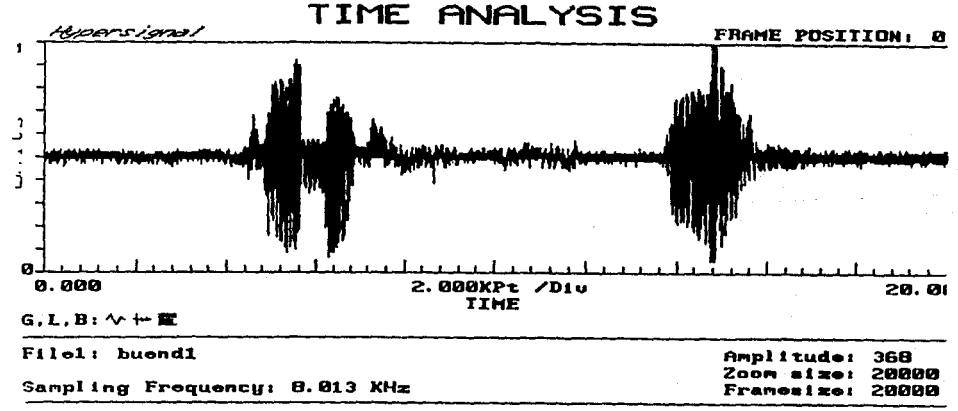
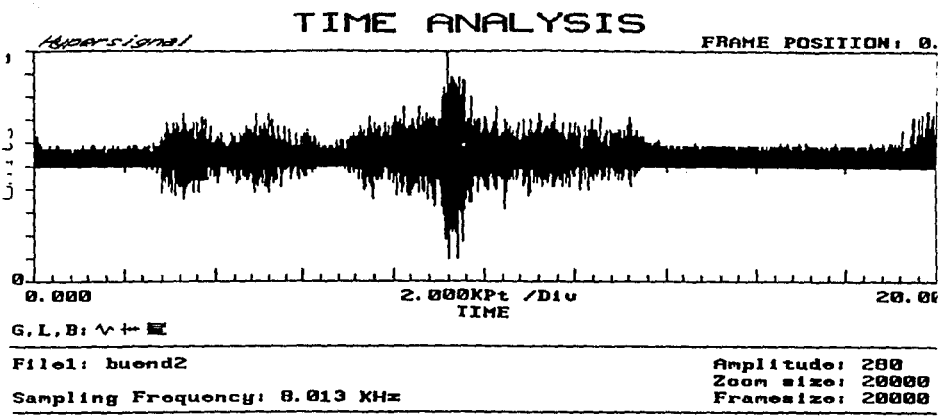


Figura I.7 Formas de onda de la palabra "sabio"


TIME ANALYSIS
Hypersignal
FRAME POSITION: 0
0
Units
.0
0.000
2.000KPt /Div
TIME
20.0
G,L,B:
File1: buendia
Sampling Frequency: 8.013 KHz
Amplitude: 568
Zoom size: 20000
Framesize: 20000


TIME ANALYSIS
Hypersignal
FRAME POSITION: 0
1
0
0.000
2.000KPt /Div
TIME
20.0
G,L,B:
File1: buend1
Sampling Frequency: 8.013 KHz
Amplitude: 368
Zoom size: 20000
Framesize: 20000



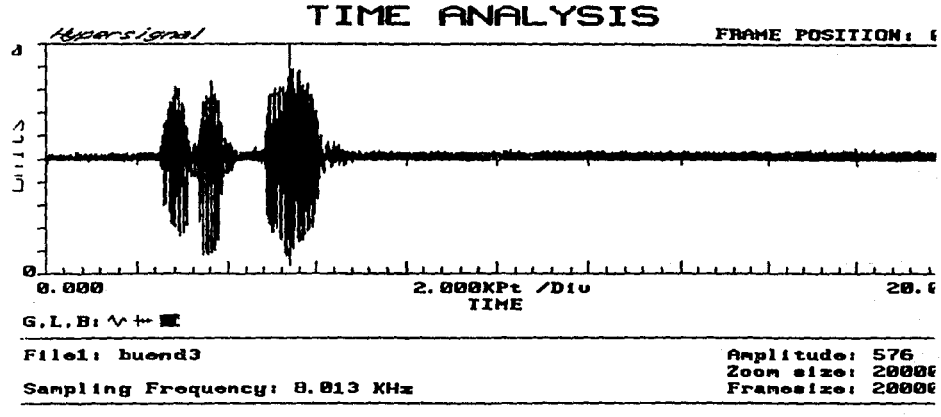


Figura I.8 Formas de onda de las palabras "Buenos Dias"

II REPRESENTACION DE UNA SEÑAL MEDIANTE FUNCIONES ORTOGONALES

II.1 SEÑALES ORTOGONALES Y EL PROCEDIMIENTO DE GRAM-SCHMIDT

En las últimas décadas ha habido un interés creciente respecto al estudio de la representación de señales en función de otras señales ortogonales en el área del procesamiento digital. La más conocida en este campo es la que utiliza la transformada de Fourier. La búsqueda de mayores aplicaciones utilizando esta transformada ha llevado a los estudiosos del tema a implementar la transformada discreta de Fourier (DFT) y la transformada rápida de Fourier (FFT). Sin embargo, existen algunas otras como pueden ser la de Walsh y Haar que se estudiarán más adelante. La gran cantidad de avances que se han logrado se debe en parte al aumento de velocidad y capacidad de procesamiento de las computadoras. Esto se ha visto reflejado en dispositivos digitales con propósitos específicos, así como en el desarrollo de procesadores especiales para el procesamiento digital de señales.

Los temas de investigación en donde se han aprovechado las propiedades de las señales ortogonales son principalmente: procesamiento de imagen, procesamiento de voz, reconocimiento de patrones, análisis y diseño de sistemas de comunicación, así como filtrado de señales.

II.1.1 DEFINICION DE LAS SEÑALES ORTOGONALES

Si x(t) representa una señal (análogica) que es continua respecto al tiempo t, y que es la entrada de un convertidor ideal análogico-digital que muestrea a razón de N muestras por segundo, entonces la salida del convertidor es una señal x*(t) discreta (o muestreada), definida por

$$x^*(t) = \Delta t \sum_{m=0}^{N-1} x(m\Delta t)\ \delta\ (t - m\Delta t) \qquad (II.1)$$

En la ecuación II.1.1.1, Δt es el intervalo de muestreo, y $\delta(t)$ representa la función delta de Dirac.

El muestreo anterior resulta en la secuencia de datos, $\{X(m)\}$, $m = 0,1, \ldots N-1$, en donde $x(m)=x(m\Delta t)$. El término secuencia digital implica que cada X(m) debe ser cuantificado y codificado de manera digital.

II.1.2 REPRESENTACION DE LAS SEÑALES UTILIZANDO SEÑALES ORTOGONALES

Basado en las observaciones realizadas sobre un fenómeno en particular, un investigador puede proponer un modelo matemático a partir del cual se hará un análisis subsecuente. En muchas aplicaciones este modelo toma la forma de una combinación lineal de un conjunto de señales especificadas como $\{Un(t)\}$.

Sea un conjunto de funciones continuas y reales: $\{Un(t)\} = \{Uo(t), U_1(t), \ldots\}$. Para que el conjunto de funciones definidas por {Un (t)} sea ortogonal en el intervalo (to, to + T), es necesario, que:

$$\int_T Um(t)\, Un(t)\, dt = \begin{cases} c_m \text{ si } m = n \\ 0 \text{ si } m \neq n \end{cases} \qquad \text{(II.2)}$$

En donde la notación :

$$\int_T \text{ significa: } \int_{to}^{to+T}$$

En el caso de que $c_m = 1$, {Un (t)} se dice que es un conjunto de señales ortonormales. Si cm es diferente de 1 y se quiere normalizar las señales, es necesario tomar los anteriores Um(t) y dividirlos por $(c_m)^{1/2}$ para formar las nuevas señales Um(t) ya normalizadas.

Suponga que x(t) es una señal de valores reales definidos en el intervalo (to, to + T), entonces ésta se representa por la expansión:

$$x(t) = \sum_{n=0}^{\infty} am\ Un\ (t) \qquad \text{(II.3)}$$

En donde am representa el n-ésimo coeficiente de la expansión y está dado por:

$$am = \frac{1}{c_m} \int_T x(t)\, Um(t)\, dt \quad n = 0, 1 \ldots \qquad \text{(II.4)}$$

Para entender lo anterior, si se cuenta con un conjunto de señales continuas y reales {Un(t)}. Utilizando la ecuación II.3, se aplica en ambos lados de ésta el siguiente operador de integración:

$$\int_T [\text{-}]\, Um(t)\, dt$$

se obtiene:

$$\int_T x(t)\, Um(t)\, dt = \int_T \sum_{n=0}^{\infty} am\, Un(t) Um(t)\, dt$$

que es equivalente a decir

$$\Leftrightarrow \quad \int_T x(t)\, Um(t)\, dt = \sum_{n=0}^{\infty} am \int_T Un(t)\, Um(t)\, dt$$

$$\Leftrightarrow \quad \int_T x(t)\, Um(t)\, dt = \sum_{n=0}^{\infty} am\, Cm$$

ya que

$$\int_T Un(t)\, Um(t) = 0$$

si $m \neq n$ según la ecuación II.2, entonces

$$\Leftrightarrow \quad \int_T x(t)\, Um(t)\, dt = am\, c_m$$

despejando se obtiene:

$$am = \frac{1}{c_m} \int_T x(t)\, Um(t)\, dt \quad n = 0, 1 \ldots.$$

con lo que se demuestra la ecuación II.4

Ahora se define un conjunto ortogonal completo o cerrado. Sea {Un(t)} un conjunto de funciones continuas y reales que verifiquen

a) $$\sum_{\substack{n=0 \\ n\neq m}}^{\infty} \sum_{\substack{m=0 \\ n\neq m}}^{\infty} \int_T Um(t)\, Un(t)\, dt = 0 \qquad (II.5)$$

Es decir, es un conjunto de señales ortogonales.

b)

$$\int_T U^2n(t)\, dt < \infty \qquad (II.6)$$

Es decir, cada señal posee una energía finita.

El conjunto {Un(t) } se dice ser completo o cerrado si se cumple cualquiera de los siguientes postulados:

1) No existe alguna señal x (t) con

$$\int_T x^2(t)\, dt < \infty$$

tal que

$$\int_T x(t)\, Un(t)\, dt = 0 \qquad n = 0, 1 \ldots. \quad (II.7)$$

2) Para cada señal contínua x (t) con

$$\int_T x^2(t)\, dt < \infty$$

y un número ε, $\varepsilon > 0$, pequeño. Existe un número entero N y una expansión finita

$$x'(t) = \sum_{n=0}^{N-1} an\, Un(t) \qquad (II.8)$$

de tal manera que

$$\int_T | x(t) - x'(t) |^2 \, dt < \varepsilon \qquad (II.9)$$

$$\frac{1}{T} \sum_{n=0}^{N-1} C_n a^2_n < \infty \qquad (II.10)$$

x'(t) es la señal modelada que hemos obtenido en base al conjunto de funciones Un(t). En el caso ideal, x'(t) debe ser idéntico a x(t); sin embargo, es posible que con {Un(t)} no se forme una combinación lineal perfecta y exista un cierto error.

La ecuación II.9 da una medida de este error, y por lo tanto, expresa la calidad del modelo propuesto.

Si

$$\int_T | x(t) - x'(t) |^2 \, dt$$

es muy grande, debemos buscar otro modelo seleccionando un conjunto de señales diferente a Un (t).

La ecuación II.10 implica que la señal modelada x'(t) debe tener una energía finita basado en los coeficientes an con los que se define la señal en {Un(t)}. Para comprender mejor ésto, se toma la ecuación II.3 para el caso de n=0,1; es decir,

$$x(t) = \sum_{n=0}^{1} an\, Un(t).$$

Si cada lado de la igualdad se eleva al cuadrado, se obtiene:

$$x^2(t) = \left(\sum_{n=0}^{1} an\, Un(t) \right)^2$$

$$x^2(t) = (ao\ Uo(t) + a_1 U_1(t))^2$$

$$x^2(t) = a_o^2 U_o^2(t) + a_1^2 U_1^2(t) + a_o U_o(t)\, a_1 U_1(t) + a_1 U_1(t)\, a_o U_o(t)$$

La última ecuación se puede expresar como:

$$x^2(t) = \sum_{n=0}^{1} a^2 n\, U^2 n(t) + \sum_{p=0}^{1} \sum_{\substack{q=0 \\ p \neq q}}^{1} ap\, Up(t)\, aq\, Uq(t)$$

al generalizar para n = 0, 1 ...

$$x^2(t) = \sum_{n=0}^{\infty} a^2 n\, U^2 n(t) + \sum_{p=0}^{\infty} \sum_{\substack{q=0 \\ p \neq q}}^{\infty} ap\, Up(t)\, aq\, Uq(t)$$

al integrar en ambos lados de la igualdad se obtiene:

$$\int_T x^2(t)\,dt = \int_T \left[\sum_{n=0}^{\infty} a^2 n\, U^2 n(t) + \sum_{p=0}^{\infty} \sum_{\substack{q=0 \\ p \neq q}}^{\infty} ap\, Up(t)\, aq\, Uq(t) \right] dt$$

$$\Leftrightarrow \int_T x^2(t)\,dt = \int_T \sum_{n=0}^{\infty} a^2 n\, U^2 n(t)\,dt + \int_T \sum_{p=0}^{\infty} \sum_{\substack{q=0 \\ p \neq q}}^{\infty} ap\, Up(t)\, aq\, Uq(t)\,dt$$

$$\Leftrightarrow \int_T x^2(t)\,dt = \sum_{n=0}^{\infty} a^2 n \int_T U^2 n(t)$$

$$+ \sum_{\substack{p=0 \\ p \neq q}}^{\infty} \sum_{q=0}^{\infty} ap\, aq \int_T Up(t)\, Uq(t)\, dt$$

$$\int_T Up(t)\, Uq(t)\, dt = 0 \quad ya \quad que \quad p \neq q$$

$$\Leftrightarrow \quad \int_T x^2(t)\, dt = \sum_{n=0}^{\infty} a^2 n \int_T U^2 n(t)\, dt$$

$$\Leftrightarrow \quad \int_T x^2(t)\, dt = \sum_{n=0}^{\infty} a^2 n\, Cn$$

$$\Leftrightarrow \quad \int_T x^2(t)\, dt = \sum_{n=0}^{\infty} Cn\, a^2 n$$

Dividiendo por T en ambos miembros,

$$\frac{1}{T} \int_T x^2(t)\, dt = \frac{1}{T} \sum_{n=0}^{\infty} Cn\, a^2 n \qquad (II.11)$$

El resultado de la ecuación II.11 es conocido como el *teorema de Parseval*. Si x(t) es el voltaje de una señal conectada a los bornes de una resistencia de un ohm, entonces la ecuación II.11 representa la potencia promedio disipada por la resistencia; y el conjunto de valores $\{Cn\, a^2 n/T\}$ la distribución de potencia en x (t).

II.1.3 ESPACIO VECTORIAL N-DIMENSIONAL

Supongamos ahora que se tiene una señal s(t) de valores reales y continua en el tiempo. Si existe un conjunto de señales ortogonales { Un(t) }, entonces de acuerdo con la ecuación II.3 se puede expresar:

$$s(t) = \sum_{n=0}^{N-1} an\, Un(t)$$

siempre y cuando { Un (t) } sea cerrado.

La señal s(t) puede ser generada utilizando un número razonable de señales ortogonales Un(t). Como se observa en la figura II.1 s(t) se puede sintetizar sumando las funciones Un(t) con sus respectivos pesos específicos dados por {an}.

Una manera para representar s(t) diferente a la síntesis mediante las funciones {Un(t)}, es asociando un espacio vectorial N-dimensional. Si s(t) puede ser representado por el conjunto de valores {an} entonces se puede decir que {Un(t)} puede generar un espacio vectorial donde la dimensión estará dada por n

$$s = (a_0, a_1, \ldots a_{n-1})$$

es un vector de la señal s(t) cuyos elementos a_n están dados por:

$$a_n = \frac{1}{Cn} \int_T s(t)\, Un(t) \qquad n = 0, 1, \ldots$$

Supongamos que para representar s(t) son necesarias tres señales ortogonales {Un (t)} = { U_0 (t), U_1 (t), U_2 (t)}. En este espacio vectorial s(t) estará dado por:

$$s(t) = [\, U_0\, U_1\, U_2] \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix}$$

$$s = \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix}$$ s es el vector de coeficientes que da la representación de S(t) en {Un(t)}

Una representación gráfica podría ser la que se muestra en la figura II.2:

Esta misma base ortogonal puede ser utilizada para representar hasta m señales $s_m(t)$
m = 0,1, ...

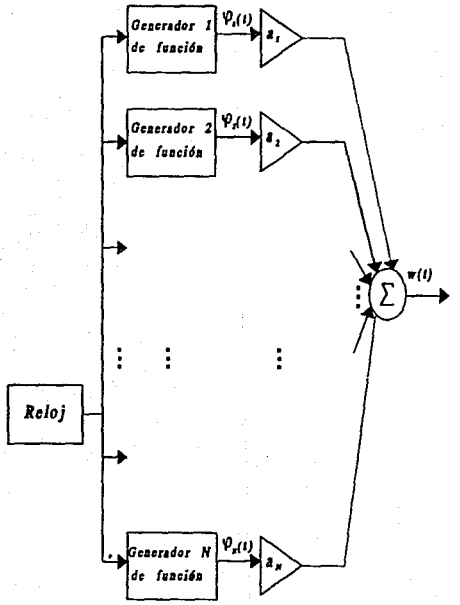


Figura II.1 Síntesis de la señal utilizando funciones ortogonales.

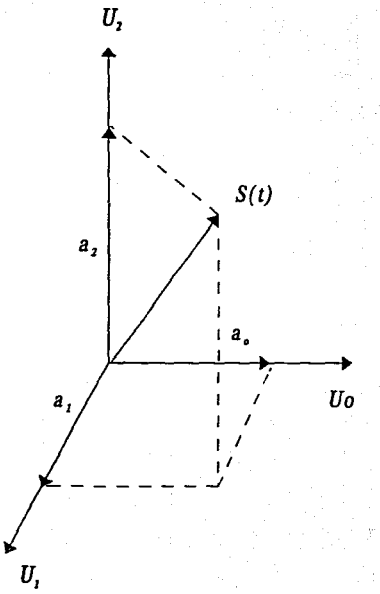


Figura II.2 Representación gráfica para S(t)

La representación vectorial de $s_m(t)$ puede variar si se aumenta el número de señales base {Un(t)} o si se reemplaza por otro conjunto, {φ n (t)} por ejemplo:

Dos tipos diferentes de forma de onda de señal se pueden seleccionar para formar una base:

1) Señales formadas por división de frecuencia
2) Señales formadas por división de tiempo
3) Señales especiales

II.1.4 SEÑALES FORMADAS POR DIVISION DE FRECUENCIA

Si un conjunto de funciones {Un (t)} está limitado en tiempo al intervalo (O, To) y no existe traslape en las frecuencias de cada una de las señales, entonces se puede decir que {Un (t)} es un conjunto ortonormal.

Un ejemplo sencillo de lo anterior es el conjunto de pulsos senoidales que se ilustra en la figura II.3

Este es un conjunto de señales ortonormales dado por:

$$Un(t) = \begin{cases} (2/To)^{1/2} \sin [2\pi nt/To] & 0 < t \leq To \\ 0 \quad \text{de otra manera} \end{cases}$$

en donde n = 0, 1, ...

Es claro que para cualquier par de funciones senoidales tales que

$$f1 = \sin nwt$$
$$f2 = \sin mwt \qquad n \neq w$$

$$\int_T f1\ f2\ dt = \int_T \sin(nwt) \sin(mwt)\ dt$$

recordando de las fórmulas de Euler que $e^{jwt} = \sin wt$

$$\Leftrightarrow \quad \int_T f1\ f2\ dt = \int_T e^{jnwt}\ e^{jmwt}\ dt$$

$$\Leftrightarrow \quad \int_T f1\ f2\ dt = \int_T e^{jwt\,(n+m)}\ dt$$

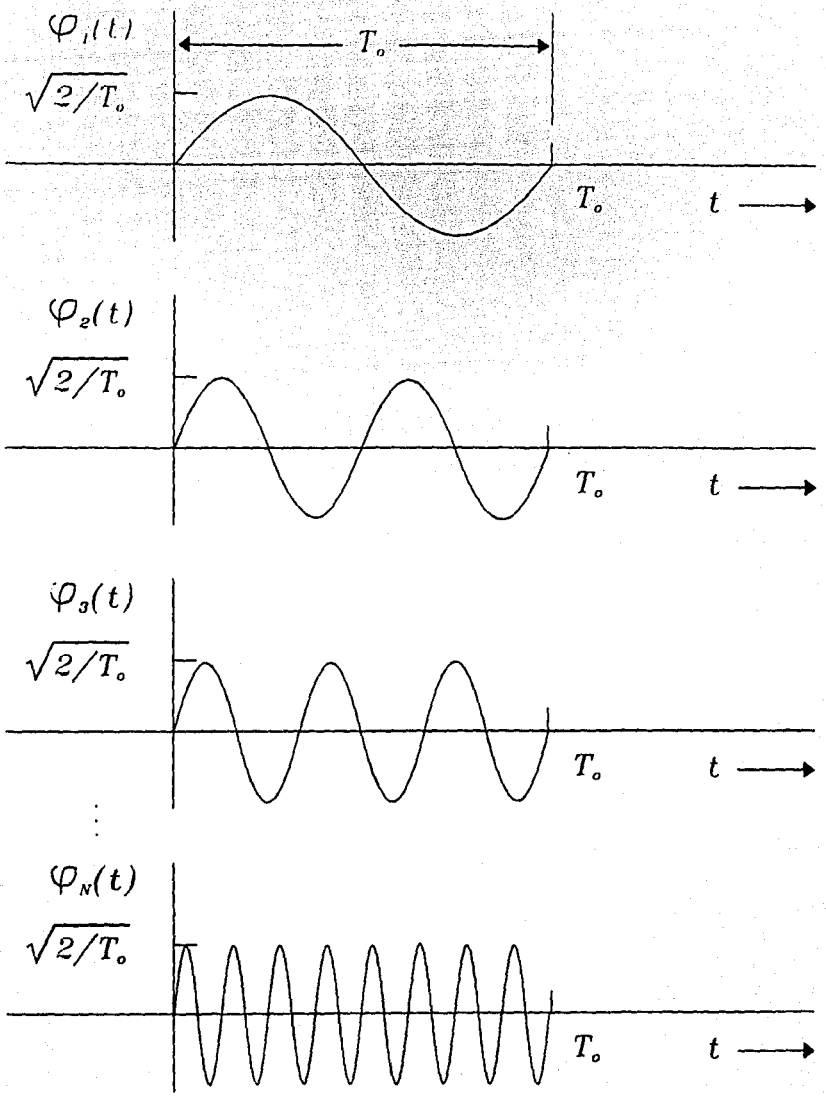


Figura II.3 Conjunto de pulsos senoidales

$$\Leftrightarrow \quad \int_T f1\ f2\ dt = \int_T \sin [\ (n + m)\ wt\ ]\ dt$$

$$\Leftrightarrow \quad \int_T f1\ f2\ dt = 0$$

De la misma manera que es posible crear un conjunto de señales ortogonales con las funciones senoidales y teniendo como argumentos frecuencias múltiples de una frecuencia fundamental, también es posible hacerlo con funciones cosenoidales.

El caso más completo es aquél en que se consideran tanto funciones senoidales como cosenoidales. A este tipo de modelado se le llama: *Representación por series de Fourier.*

II.1.5 REPRESENTACION POR SERIES DE FOURIER

Se considera el caso en que se tiene un conjunto de funciones {Un (t)}; si este conjunto de funciones está dado por {1, cos nw_0t, sin nw_0t} con n = 0, 1, ... entonces la serie que corresponde a la ecuación II.3 es

$$x(t) = a_0 + \sum_{n=1}^{\infty} a_n \cos n w_0 t + \sum_{n=1}^{\infty} b_n \sin w_0 t \qquad (II.12)$$

donde, w_0 (radianes por segundo) es la frecuencia fundamental angular que se encuentra relacionada al período T (en segundos) por la fórmula $T=2\pi/w_0$. La frecuencia fundamental angular es igual a 2π veces la frecuencia fundamental f_0 (ciclos por segundo o Hertz). Las frecuencias nw_0 o nf_0 son llamadas Armónicas puesto que son múltiplos enteros de las frecuencias fundamentales w_0 y f_0 respectivamente. Tomando en cuenta que

$$\int_T x^2\ dt < \infty$$

Se puede decir que las series en la ecuación E.F convergen en el intervalo (0,T) y por lo tanto $\{a_n\}$ y $\{b_n\}$ son finitos.

El conjunto de funciones {cos nw_0t, sin nw_0t} es ortogonal. De aquí es posible verificar las siguientes propiedades:

$$\int_T \cos n w_0 t \quad \cos m w_0 t\ dt = \begin{cases} T/2 & m = n \\ 0 & m \neq n \end{cases}$$

$$\int_T \cos n w_o t \quad \sin m w_o t \, dt = 0 \; \forall \; m, n$$

$$\int_T \sin n w_o t \quad \sin m w_o t \, dt = \begin{cases} T/2 & m = n \\ 0 & m \neq n \end{cases}$$

Ahora, se busca determinar los coeficientes a_o, a_n, b_n. Tomando la ecuación

$$x(t) = a_o + \sum_{n=1}^{\infty} a_n \cos(n w_o t) + \sum_{n=1}^{\infty} b_n \sin(n w_o t)$$

al integrar

$$\int_T [\cdot] \cos_m w_o t$$

$$\int_T x(t) \cos_m w_o t \, dt = \int_T a_o \cos_m w_o t \, dt + \int_T \sum_{n=1}^{\infty} a_n \cos_n w_o t \cos_m w_o t \, dt + \int_T \sum_{n=1}^{\infty} b_n \sin_n w_o t \cos_m w_o t \, dt$$

$$\Leftrightarrow \int_T x(t) \cos_m w_o t \, dt = a_o \int_T \cos_m w_o t \, dt + \sum_{n=1}^{\infty} a_n \int_T \cos_n w_o t \cos_m w_o t \, dt + \sum_{n=1}^{\infty} b_n \int_T \sin_n w_o t \cos_m w_o t \, dt$$

$$\Leftrightarrow \quad \int_T x(t)\ \cos_m w_o t\ dt = a_m \frac{T}{2}$$

$$\Leftrightarrow \quad a_m = \frac{2}{T} \int_T x(t)\ \cos_m w_o t\ dt = a_o$$

m = 1, 2, ...

Si la ecuación se multiplica por:

$$\int_T [.]\ \sin_m w_o t\ dt$$

$$\int_T x(t)\ \sin_m w_o t\ dt = \int_T a_o \sin_m w_o t\ dt + \int_T \sum_{n=1}^{\infty} a_n\ \cos_n w_o t\ \sin_m w_o t\ dt + \int_T \sum_{n=1}^{\infty} b_n\ \sin_n w_o t\ \sin_m w_o t\ dt$$

$$\Leftrightarrow \quad \int_T x(t)\ \sin_m w_o t\ dt = a_o \int_T \sin_m w_o t\ dt + \sum_{n=1}^{\infty} a_n \int_T \cos_n w_o t\ \sin_m w_o t\ dt + \sum_{n=1}^{\infty} b_n \int_T \sin_n w_o t\ \sin_m w_o t\ dt$$

$$\Leftrightarrow \quad \int_T x(t)\ \sin_m w_o t\ dt = b_m T/2$$ despejando bm:

$$b_m = 2/T \int_T x(t)\ \sin_m w_o t\ dt \qquad m = 1, 2, \ldots$$

finalmente, la ecuación II.12 se multiplica por $\int_T [.]\, dt$

$$\int_T x(t)\, dt = \int_T a_o\, dt + \int_T \sum_{n=1}^{\infty} a_n \cos_n w_o t\, dt + \int_T \sum_{n=1}^{\infty} b_n \sin_n w_o t\, dt$$

$$\Leftrightarrow \quad \int_T x(t)\, dt = a_o \int_T dt + \sum_{n=1}^{\infty} a_n \int_T \cos(n w_o t\, dt) + b_n \int_T \sin(n w_o t\, dt)$$

$$\Leftrightarrow \quad \int_T x(t)\, dt = a_o T \qquad \text{despejando } a_o$$

$$a_o = \frac{1}{T} \int_T x(t)\, dt$$

Resumiendo se tiene:

$$ao = \frac{1}{T} \int_T x(t)\, dt$$

$$an = \frac{2}{T} \int_T x(t) \cos(n\ wot\ dt)$$

$$bn = \frac{2}{T} \int_T x(t) \sin(n\ wot\ dt)$$

$$n = 1, 2, \ldots$$

De la discusión anterior es posible concluir que la señal x(t) se puede representar por el conjunto de números reales $\{a_o, a_n, b_n\}$. Para el caso más sencillo en que $n=1$, $w_o=1$, se considera la función x(t) que se muestra en la figura II.4 .

Como $n=1$ el conjunto de funciones ortogonales estará formado por $\{Un(t)\} = \{1, \cos t, \sin t\}$. Lr los coeficientes ao, a1, b1 se pueden encontrar como:

$$a_o = \frac{1}{T}\int_T x(t)\,dt = \frac{1}{4\pi}\int_\pi^{2\pi} 2\,dt = 1/2$$

$$a_1 = 2/T\int_T x(t)\cos t\,dt = 1/2\pi\int_\pi^{2\pi} 2\cos t\,dt \quad a_1 = [\sin t]_\pi^{2\pi} = 0$$

$$b_1 = 2/T\int_T x(t)\sin t\,dt = 1/2\pi\int_\pi^{2\pi} 2\sin t\,dt = [-\cos t/p]_\pi^{2\pi} = (-1/\pi - 1/\pi) = -1/\pi$$

$$x(t) = [\,1, \cos t, \sin t\,]\begin{bmatrix} 0.5 \\ 1 \\ -1/\pi \end{bmatrix}$$

El vector de la señal x es

$$x = \begin{bmatrix} 0.5 \\ 1 \\ -1/\pi \end{bmatrix}$$

y una representación gráfica sería la que se observa el la figura II.5:

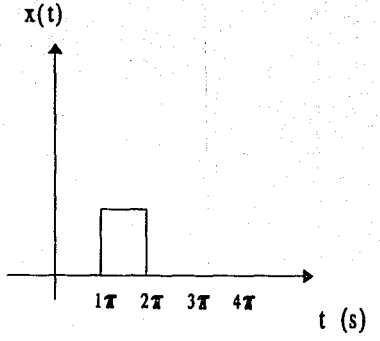


Figura II.4 Funcion x(t)

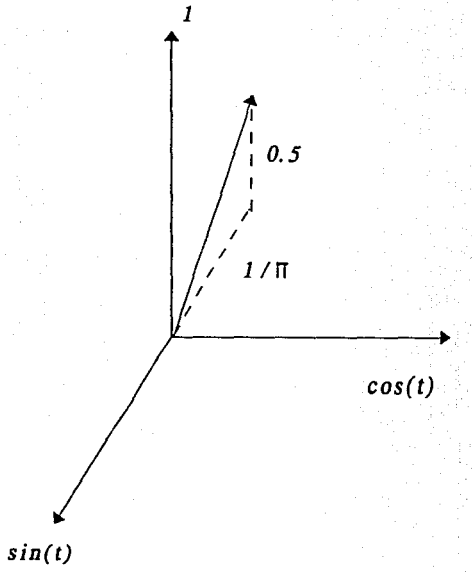


Figura II.5 Representacion para x(t) en Un

En este caso n=1 es demasiado pequeño y es claro que debe existir un error muy grande entre x(t) y el modelado $\hat{x}(t)$. Sin embargo, para fines demostrativos se prefiere trabajar con el caso más sencillo.

II.1.6 SEÑALES POR DIVISION DE TIEMPO

Es posible crear un conjunto de señales ortogonales {Un (t)} si se tienen N pulsos que no se traslapen y que la suma de todos ellos no sea cero a lo largo de todo el intervalo (0, T_0) y cada función Un (t) no sea cero en el intervalo T_0/N.

El conjunto de pulsos rectangulares que se ilustra en la figura II.6 es un ejemplo

Este conjunto de funciones está expresado por:

$$\text{Un (t)} = \begin{cases} 1/(T)^{1/2} & (n-1)\ T < t \leq nT \\ 0 & \text{de otra manera} \end{cases}$$

donde To = NT y n=1, 2, ... N. En este caso nunca hay traslape de Un (t) y Um (t) para m≠n.

Hay que hacer notar que no todas las funciones tienen que encontrarse dentro de esta clasificación para poder ser ortogonales.

II.1.7 SEÑALES ESPECIALES

En esta sección se introduce el tipo de funciones que no son senoidales, que tampoco pueden definirse por división en tiempo ya que puede existir un traslape entre los diferentes Un (t), pero que sin embargo son ortogonales. Estas señales son:

1) Rademacher
2) Haar
3) Walsh

Las funciones ortogonales mencionadas consisten de ondas cuadradas o rectangulares. Las señales individuales que pertenecen a estos conjuntos se distinguen por medio de un parámetro llamado **secuencia.**

El término de frecuencia se aplica a un conjunto de funciones senoidales periódicas teniendo un intervalo de tiempo de cruce-por-cero uniforme. Este es el parámetro f que distingue a las funciones individuales que pertenecen a los conjuntos {cos 2π ft} y {sin 2π ft} y se interpreta como el número de ciclos completos generados por una señal senoidal por unidad de tiempo.

La generalización de la frecuencia se conoce como frecuencia generalizada y es la mitad del promedio de cruces por cero en una unidad de tiempo. Haramuth introdujo el término de secuencia para describir la frecuencia generalizada y aplicarla para distinguir funciones uniformemente espaciadas en un intervalo de tiempo y que no son necesariamente periódicas. La definición de secuencia coincide con aquella frecuencia cuando se aplica a funciones senoidales.

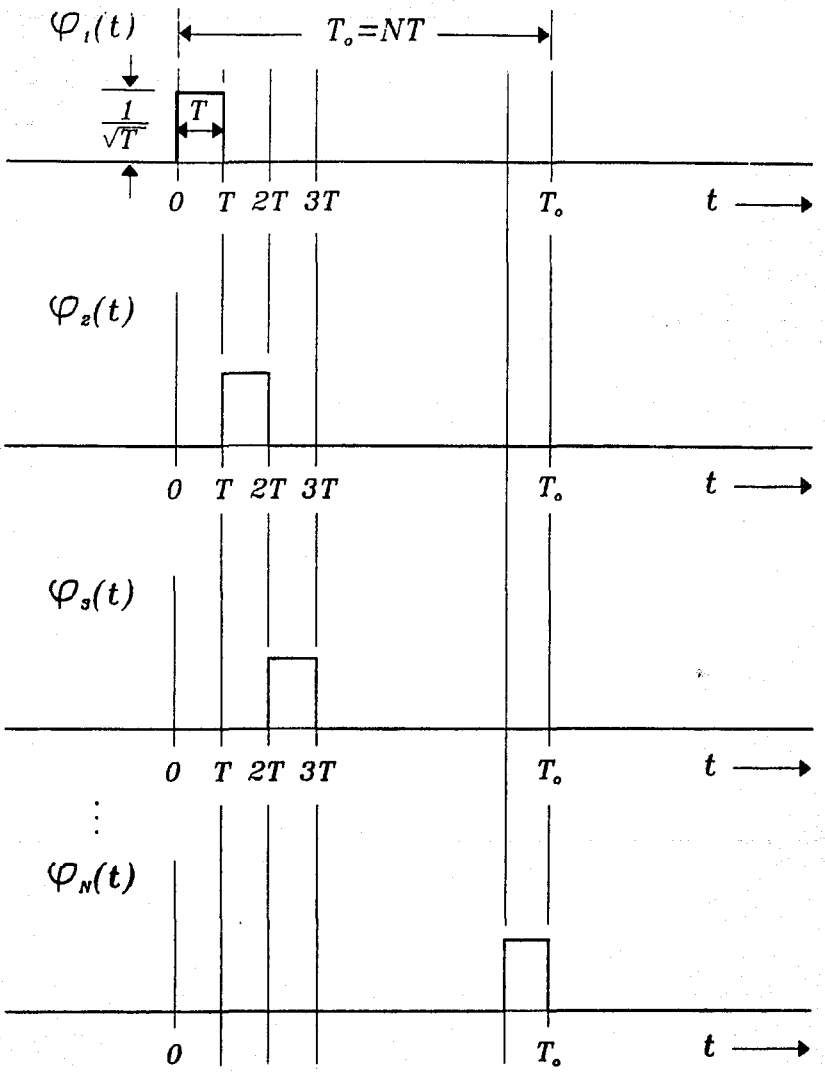


Figura II.6 Conjunto de pulsos rectangulares

Aplicando la definición anterior para funciones periódicas y aperiódicas se obtiene:

1) La secuencia de una función periódica es igual a la mitad del número de cambios de signo en un período.

2) La secuencia de una función aperiódica es igual al número de cambios de signo por unidad de tiempo, si este límite existe.

Para obtener esto, se presentan los ejemplos de la figura II.7 .

Puesto que cada función tiene cuatro cruces por cero en el intervalo de tiempo, la secuencia de cada uno de ellos es dos. De manera análoga a la frecuencia que es expresada en ciclos por segundo o en Hertz, la secuencia se expresa en términos de cruces por cero por segundo y que tendrá por abreviación "ccp".

La definición anterior de secuencia se puede aplicar con una modificación mínima a las funciones discretas. Si se tienen las funciones que se muestran en la figura II.8 .

Entonces podemos definir η como el número de cambios de signo por unidad de tiempo. Si η es par, la secuencia estará dada or $\eta/2$ y $(\eta + 1)/2$ si es impar. Para el ejemplo en ambos casos la secuencia es 2.

Para el estudio de las funciones es necesario establecer una notación con la cual se identifican. Las abreviaciones que se usarán se aprecian en la tabla II.1.

TABLA II. 1 Abreviaciones para funciones continuas y discretas

Nombre de la función	Abreviación	
	Función Continua	Función Discreta
Rademacher	rad	Rad
Haar	har	Har
Walsh	wal	Wal
"coseno de Walsh"	cal	Cal
"seno de Walsh"	sal	Sal

II.1.8 LAS FUNCIONES DE RADEMACHER Y DE HAAR

Las funciones de Rademacher son un conjunto incompleto de señales ortonormales, las cuales fueron desarrolladas en 1922. La función de Rademacher de índice m, y que se denota como rad (m,t), es un tren de pulsos rectangulares con 2^{m-1} ciclos en el intervalo abierto (0,1) siendo m = 1, 2 ... Una excepción es rad(0,t) el cual es una constante en 1.

En la figura II.9 se pueden observar las funciones de Rademacher hasta m=4.

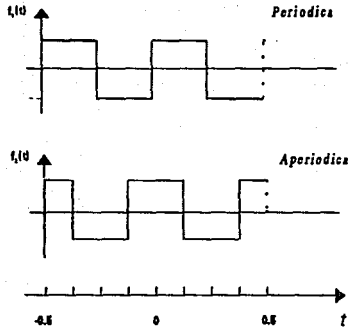


Figura II.7 Dos ejemplos de Secuencia

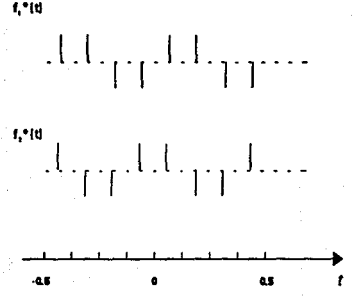


Figura II.8 Dos ejemplos de Secuencia en funciones discretas

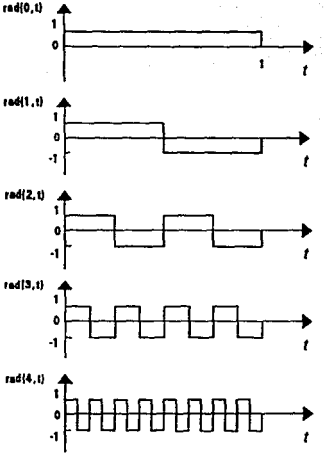


Figura II.9 Funciones de Rademacher hasta m= 4

Las funciones de Rademacher son periódicas con período 1, es decir,

$$rad\ (m,t) = rad\ (m, t + 1)$$

las funciones de Rademacher pueden ser generadas utilizando la ecuación de recurrencia

$$rad\ (m,t) = rad\ (1, 2^{m-1}\ t)$$

$$rad\ (1,t) = \begin{cases} 1 & t \in [0, 1/2] \\ -1 & t \in [1/2, 1] \end{cases}$$

El conjunto de funciones Haar {har (n, m, t)} es periódico, ortonormal y completo. Fue propuesto en 1910 por Haar. La figura II.10 muestra el conjunto de las primeras ocho funciones de Haar.

Una relación de recurrencia permite generar {har (n, m, t)}, esta es:

$$har\ (0, 0, t) = 1 \qquad t \in [0,1]$$

$$har\ (r, m, t) = \begin{cases} 2^{r/2}, & (m-1)/2r \le t < (m - 1/2)/2r \\ -2^{r/2}, & (m-1/2)/2r \le t < m/2r \\ 0 & \text{de otra manera} \end{cases}$$

donde $0 \le r < \log_2 N$ y $1 \le m \le 2^T$ donde $n = \log_2 N$

II.1.9 LAS FUNCIONES DE WALSH

El conjunto incompleto de Rademacher fue completado por Walsh en 1923 para formar el conjunto ortonormal completo de funciones rectangulares que hoy se conocen como funciones de Walsh.

El conjunto de funciones de Walsh está generalmente clasificado en tres grupos. Estos grupos se diferencian uno de otro por el orden en el que aparecen las funciones individuales. Los tres tipos de orden son:

1) Orden de secuencia o de Walsh
2) Orden de "dyadic" o de Paley
3) Orden natural o de Hadamard

Este es el orden que fue empleado originalmente por Walsh. Las funciones de Walsh que pertenecen a este conjunto se definen como:

$$S_W = \{\ Wal_W\ (i, t),\ i = 0,1 \ldots,\ N-1\ \}$$

El subíndice "w" indica que es el orden de Walsh, i indica que se trata del i-ésimo miembro de S_W.

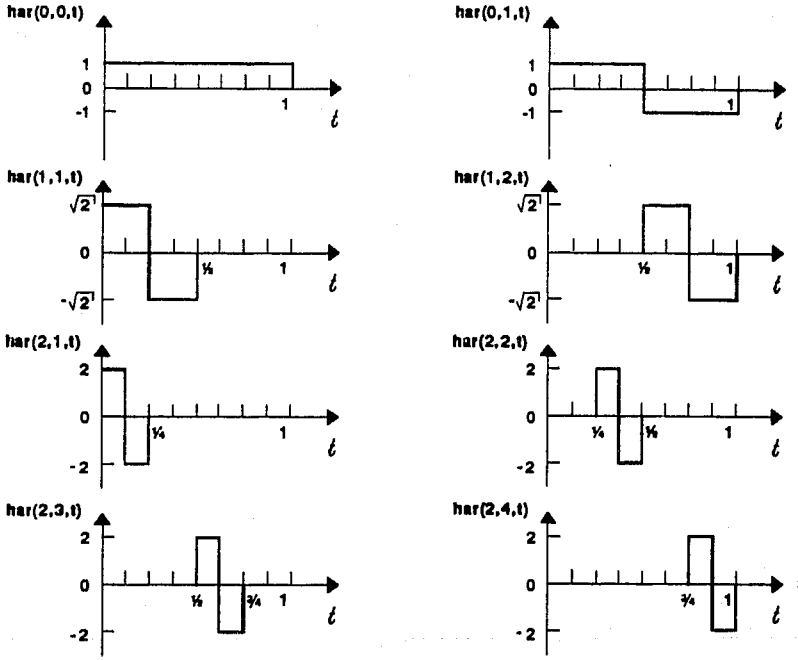


Figura II.10 Las ocho primeras funciones de Haar

Si la secuencia de Wal_W (i, t) se representa, entonces S_i esta dado por

$$S_i \begin{cases} 0 & i = 0 \\ i/2 & i = \text{par} \\ (i+1)/2 & i = \text{impar} \end{cases}$$

Cuando i es par, la función de Walsh también es una función par al igual que la función coseno, de otra manera la función se vuelve impar al igual que la función seno. Debido a esta característica tenemos:

$$\begin{aligned} &i \text{ es par } W_W(i, t) = \text{coseno Walsh} = \text{cal } (Si, t) \\ &i \text{ es impar } W_W(i, t) = \text{seno de Walsh} = \text{Sal } (Si,t) \end{aligned}$$

Las funciones de Walsh se pueden producir empleando las funciones de Rademacher y el código de Gray.

Así: para i=0 Wal_W (0,t) = rad (0,t).

Para las siguientes i-ésimas funciones de Walsh es necesario observar cuales son las funciones de Rademacher que están involucradas.

Para

i = 1 está involucrada r_1

$$Wal\ (1,t) = \int_0^t rad\ (1,t)\ dt$$

i = 2 están involucradas r_1 y r_2

$$Wal\ (2,t) = \int_0^t rad\ (1,t)\ dt\ \ rad\ (2,\ t)\ dt$$

i = 3 está involucrada r_2

$$Wal\ (3,t) = \int_0^t rad\ (2,t)\ dt$$

$i = 4$ están involucradas r_3 y r_2

$$Wal\ (4,t) = \int_0^t rad\ (3,t)\ dt\ \ rad\ (2,t)\ dt$$

$i = 5$ están involucradas r_3, r_2 y r_1

$$Wal\ (5,t) = \int_0^t rad\ (3,t)\ dt\ \ rad\ (2,t)\ \ rad(1,t)\ dt$$

.
.
.
.

En la figura II.11 se muestran las primeras funciones de Walsh.

2) Orden de "dyadic" o de Paley

El orden de dyadic fue introducido por Paley. Este conjunto de funciones de Walsh se indica

$$sp = \{ Walp\ (i, t),\ i = 0, \ldots, N\text{-}1\}$$

donde el subíndice p indica el orden de Paley e i el i-ésimo miembro de sp. El conjunto sp está relacionado con el conjunto de funciones con orden de Walsh, S_W, por la relación

$$Walp\ (i, t) = Wal_W\ [\ b\ (i),\ t]$$

donde b (i) representa la conversión de código Gray a binario.

La tabla II.2 da un resumen de la relación que existe entre las ocho primeras funciones de Sw y Sp. Este resultado se ilustra posteriormente en la figura II.12.

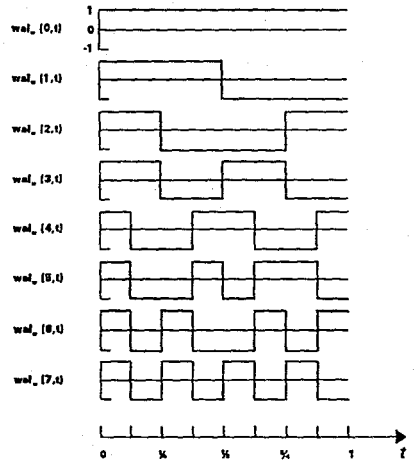


Figura II.11 Funciones de Walsh

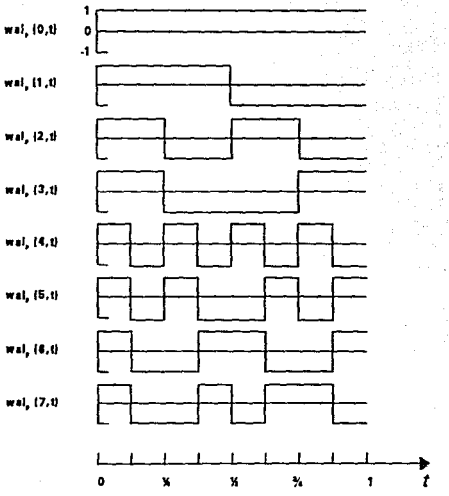


Figura II.12 Conjunto de funciones de Walsh

TABLA II.2 Relación entre Sw y Sp

i decimal	i binario	b(i) binario	b(i) decimal	$wal_p(i,t)=walw[b(i),t]$
0	000	000	0	$wal_p(0,t)=wal_w(0,t)$
1	001	001	1	$wal_p(1,t)=wal_w(1,t)$
2	010	011	3	$wal_p(2,t)=wal_w(3,t)$
3	011	010	2	$wal_p(3,t)=wal_w(2,t)$
4	100	111	7	$wal_p(4,t)=wal_w(7,t)$
5	101	110	6	$wal_p(5,t)=wal_w(6,t)$
6	110	100	4	$wal_p(6,t)=wal_w(4,t)$
7	111	101	5	$wal_p(7,t)=wal_w(5,t)$

3) El orden natural o de Hadamard

Este conjunto de funciones de Walsh es indicado por:

$$S_h = \{Wal_h\ (i, t),\ i = 0, 1, \ldots, N-1\}$$

donde el subíndice h especifica el orden de Hadamard e i indica el i-ésimo miembro de S_h.

Las funciones de S_h están relacionadas al conjunto Sw de funciones con orden de Walsh, por la relación:

$$Wal_h\ (i, t) = Wal_w\ [b(<i>), t]$$

donde $<i>$ se obtiene por el reverso de i, $b(<i>)$ es la conversión del código Gray a binario de $<i>$.

En la tabla II.3 se muestran cómo están relacionadas las ocho primeras funciones de S_w y S_h. La figura II.13 ilustra las funciones de Hadamard.

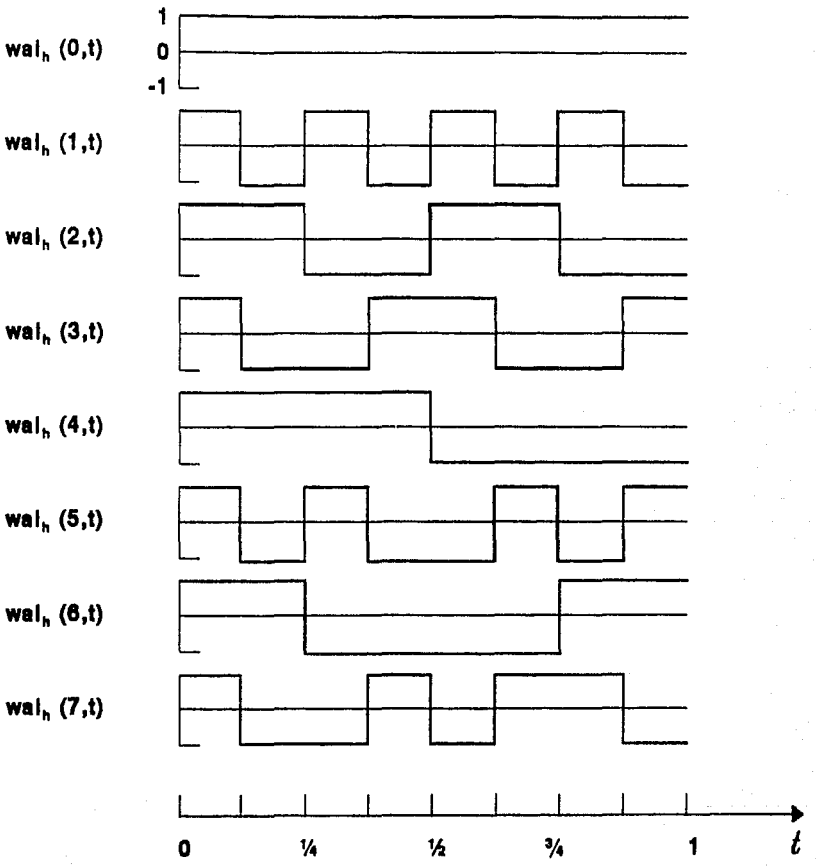


Figura II.13 Funciones de Hadamard

TABLA II.3 Relación entre Sh y Sw

i	i binario	<i> binario	b(<i>) binario	b(<i>) decimal	walh(i,t)=walw [b(<i>),t]
0	000	000	000	0	$wal_h(0,t)=wal_w(0,t)$
1	001	100	111	7	$wal_h(1,t)=wal_w(7,t)$
2	010	010	011	3	$wal_h(2,t)=wal_w(3,t)$
3	011	110	100	4	$wal_h(3,t)=wal_w(4,t)$
4	100	001	001	1	$wal_h(4,t)=wal_w(1,t)$
5	101	101	110	6	$wal_h(5,t)=wal_w(6,t)$
6	110	011	010	2	$wal_h(6,t)=wal_w(2,t)$
7	111	111	101	5	$wal_h(7,t)=wal_w(5,t)$

Como ejemplo se presentan las señales $S_1(t)$, $S_2(t)$, $S_3(t)$ y $S_4(t)$ que se muestran en la figura II.14 .

$$\text{Sea } S_1 \begin{cases} -1 & 0 \le t < 3/8 \\ 1 & 3/8 \le t < 6/8 \\ -1 & 6/8 \le t < 1 \end{cases} \qquad \begin{array}{l} 0 \le t \le 0.375 \\ 0.375 \le t < 0.750 \\ 0.75 \le t < 1 \end{array}$$

$$S_2 \begin{cases} 0 & 0 \le t < 3/8 \\ -1 & 3/8 \le t < 1 \end{cases} \qquad \begin{array}{l} 0 \le t < 0.375 \\ 0.375 \le t < 1 \end{array}$$

$$S_3 \begin{cases} 1 & 0 \le t < 6/8 \\ 3 & 6/8 \le t < 1 \end{cases} \qquad \begin{array}{l} 0 \le t < 0.75 \\ 0.75 \le t < 1 \end{array}$$

$$S_4 \begin{cases} 1 & 0 \le t < 3/8 \\ -2 & 3/8 \le t < 1 \end{cases} \qquad \begin{array}{l} 0 \le t < 0.375 \\ 0.375 \le t < 1 \end{array}$$

Como referencia, se toma el conjunto de señales ortogonales:

$$U_n(t) = \{ 1, \cos 2\pi t, \sin 2\pi t \}$$

Entonces se buscan definir los correspondientes a_0, a_1, b_1 que caracterizan a las señales en el espacio vectorial formado por {Un}; en todos los casos T=1.

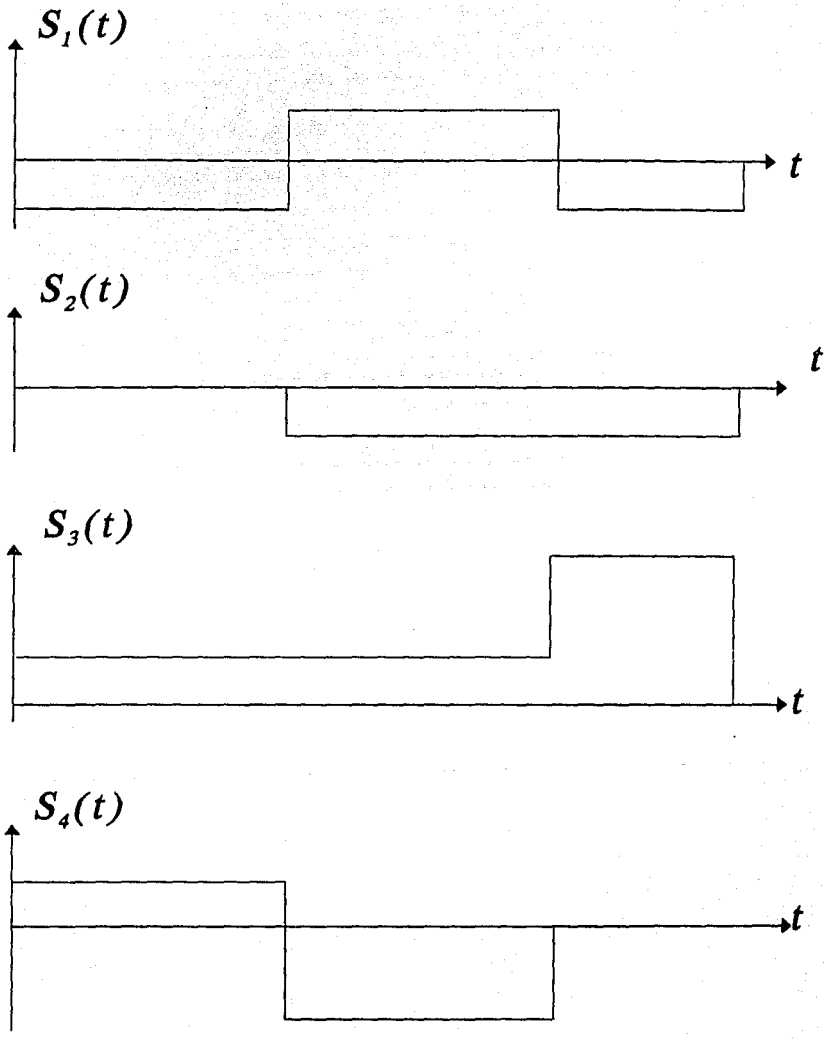


Figura II.14 Definición de las señales S_1, S_2, S_3, S_4.

Para S_1:

$$a_o = \int_T S(t)\,dt$$

$$a_o = \int_0^{3/8} -\,dt + \int_{3/8}^{6/8} dt + \int_{6/8}^{1} -\,dt$$

$$a_o = -\left[t\right]_0^{3/8} + \left[t\right]_{3/8}^{6/8} - \left[t\right]_{6/8}^{1}$$

$$a_o = -3/8 + 6/8 - 3/8 - 1 + 6/8$$

$$a_o = -2/8 = -1/4$$

$$a_1 = 2\int_T S(t)\cos 2\pi t\,dt$$

$$a_1 = 2\int_0^{3/8} -\cos 2\pi t\,dt + 2\int_{3/8}^{6/8} \cos 2\pi t\,dt + 2\int_{6/8}^{1} -\cos 2\pi t\,dt$$

$$a_1 = -2/2\pi\left[\sin 2\pi t\right]_0^{3/8} + 2/2\pi\left[\sin 2\pi t\right]_{3/8}^{6/8}$$

$$-2/2\pi\left[\sin 2\pi t\right]_{6/8}^{1}$$

$$a_1 = -1/\pi\,(0.7071) + 1/\pi\,(-1 - 0.7071) - 1/\pi\,(0 - 1)$$

$$a1 = -1.0867$$

$$b_1 = 2\int_T S(t)\sin 2\pi t\,dt$$

$$b_1 = 2\int_0^{3/8} -\sin 2\pi t\,dt + 2\int_{3/8}^{6/8} \sin 2\pi t\,dt + 2\int_{6/8}^{1} -\sin 2\pi t\,dt$$

$$b_1 = 2/2\pi \left[\cos 2\pi t\right]_0^{3/8} - 2/2\pi\left[\cos 2\pi t\right]_{3/8}^{6/8} + 2/2\pi\left[\cos 2\pi t\right]_{6/8}^{1}$$

$$b_1 = 1/\pi\,(-0.7071 - 1) - 1/\pi\,(0 - 0.7071) + (1 - 0)$$

$$b_1 = -0.4502$$

$$S_1(t) = \{1, \cos 2\pi t, \sin 2\pi t\} \begin{bmatrix} -0.250 \\ -1.086 \\ -0.450 \end{bmatrix}$$

Para S_2:

$$a_0 = \int_T S(t)\,dt$$

$$a_0 = \int_{3/8}^{1} -dt = -\left[t\right]_{3/8}^{1} = -(1 - 3/8) = -5/8$$

$$a_1 = 2\int_T S(t)\cos 2\pi t\,dt$$

$$a_1 = 2\int_{3/8}^{1} -\cos 2\pi t\,dt$$

$$a_1 = -2/2\pi \left[\sin 2\pi t\right]_{6/8}^{1}$$

$$a_1 = -1/\pi\,(0 - 0.7071) = 0.2251$$

$$b_1 = 2\int_T S(t)\,\sin 2\pi t\,dt$$

$$b_1 = 2\int_{3/8}^{1} -\sin 2\pi t\,dt = 2/2\pi \left[\cos 2\pi t\right]_{3/8}^{1}$$

$$b_1 = 1/\pi\,(1 - 0.7071) = 0.5434$$

$$S_2(t) = \{1, \cos 2\pi t, \sin 2\pi t\} \begin{bmatrix} -0.625 \\ 0.225 \\ 0.543 \end{bmatrix}$$

Para S_3:

$$a_0 = \int_T S(t)\,dt$$

$$a_0 = \int_0^{6/8} -dt + \int_{6/8}^{1} 3\,dt$$

$$a_0 = \left[t\right]_0^{6/8} + \left[t\right]_{6/8}^{1}$$

$$a_0 = 6/8 + 3 - 18/8 = 12/8 = 1.5$$

$$a_1 = 2\int_T S(t)\cos 2\pi t\,dt$$

$$a_1 = 2\int -\cos 2\pi t\,dt + 2\int_{6/8}^{1} 3\cos 2\pi t\,dt$$

$$a1 = 2/2\pi \left[\sin 2\pi t\right]_{0}^{6/8} + 6/2\pi \left[\sin 2\pi t\right]_{6/8}^{1}$$

$$a_1 = -1/\pi\,(-1 - 0) + 3/\pi\,(0 - (-1)) = 0.4317$$

$$b_1 = 2\int_{T} S(t)\,\sin 2\pi t\,dt$$

$$b_1 = 2\int_{0}^{6/8} \sin 2\pi t\,dt + 2\int_{6/8}^{1} 3\sin 2\pi t\,dt$$

$$b_1 = -2/2\pi \left[\sin 2\pi t\right]_{0}^{6/8} - 6/2\pi \left[\cos 2\pi t\right]_{6/8}^{1}$$

$$b_1 = -1/\pi\,(0 - 1) - 3/\pi\,(1 - 0)$$

$$b1 = -2/\pi = -0.6366$$

$$S_3(t) = \{1, \cos 2\pi t, \sin 2\pi t\} \begin{bmatrix} 1.500 \\ 0.431 \\ -0.636 \end{bmatrix}$$

$$S_3 = \begin{bmatrix} 1.500 \\ 0.431 \\ -0.636 \end{bmatrix}$$

Finalmente para S_4:

$$a_o = \int_{T} S(t)$$

$$a_o = \int_{0}^{3/8} dt + \int_{3/8}^{1} -2\,dt = 3/8 - 2\,(1 - 3/8)$$

$$a_o = -0.875$$

$$a_1 = 2\int_T S(t)\cos 2\pi t\, dt$$

$$a_1 = 2\int_0^{3/8} \cos 2\pi t\, dt + 2\int_{3/8}^{1} -2\cos 2\pi t\, dt$$

$$a1 = 2/2\pi\left[\sin 2\pi t\right]_0^{3/8} + 4/2\pi\left[\sin 2\pi t\right]_{6/8}^{1}$$

$$a_1 = 1/\pi\,(0.7071 - 0) - 2/\pi\,(0 - 0.7071)$$

$$a_1 = 0.7685$$

$$b_1 = 2\int_T S(t)\,dt$$

$$b_1 = 2\int_0^{3/8} \sin 2\pi t\,dt + 2\int_{3/8}^{1} -2\sin 2\pi t\,dt$$

$$b_1 = -2/2\pi\left[\cos 2\pi t\right]_0^{3/8} + 4/2\pi\left[\cos 2\pi t\right]_{6/8}^{1}$$

$$b_1 = -1/\pi\,(-0.7071 - 1) + 2/\pi\,(1 - (-0.7071))$$

$$b_1 = 1.6302$$

$$S_4(t) = \{1, \cos 2\pi t, \sin 2\pi t\} \begin{bmatrix} -0.875 \\ 0.768 \\ 1.630 \end{bmatrix}$$

La representación de estas señales en el espacio vectorial generado por $\{Un\ (t)\} = \{1, \cos\pi t, \sin 2\pi t\}$ se muestra en la figura II.15 :

Ahora se busca expresar el mismo conjunto de señales $\{S_n(t)\} = \{S_1(t), S_2(t), S_3(t), S_4(t)\}$, que se utilizaron en el ejemplo anterior, dentro de un nuevo espacio vectorial definido por:

1) $\{U_n(t)\} = \{U_1(t), U_2(t), U_3(t)\}$

Las señales que definen este espacio vectorial se muestran en la figura II.16

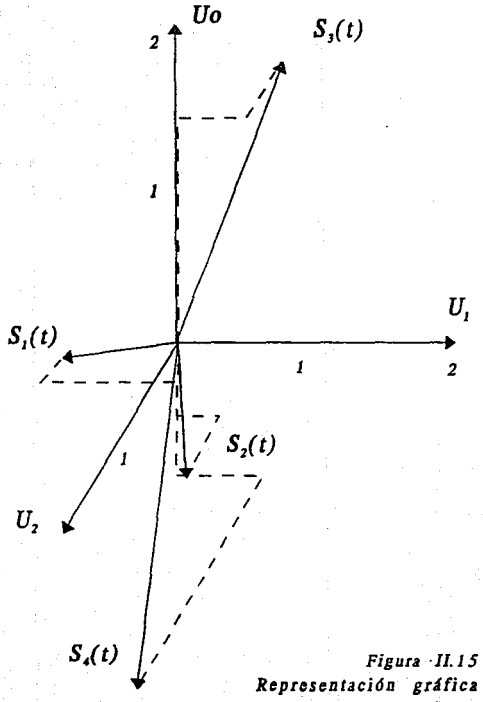


Figura II.15
Representación gráfica de S_1, S_2, S_3, S_4

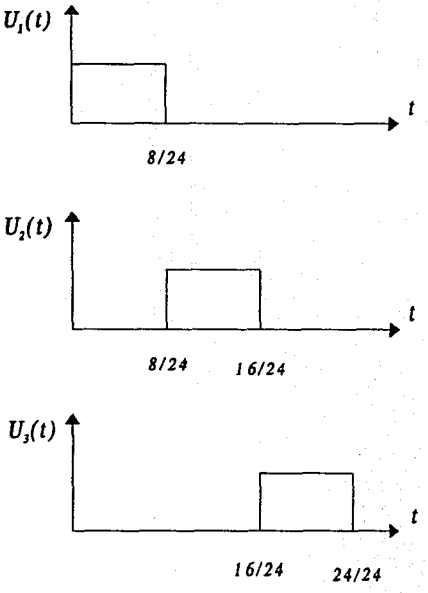


Figura II.16 Definición de las funciones U_1, U_2, U_3, U_4

De la definición:

$$a_n = 1/Cn \int_T x(t)\, U_n(t)\, dt \qquad n = 0, 1 \ldots$$

$$Cn = \int_0^{8/24} dt = 8/24$$

$$a_n = 8/24 \int_T x(t)\, U_n(t)\, dt \qquad n = 0, 1 \ldots$$

$$a_0 = 8/24 \int_0^{8/24} -\, dt = 8/24\,(-8/24) = -64/576$$

$$a_0 = -0.11$$

$$a_1 = 8/24 \int_{0.375}^{0.66} dt = 8/24\,(0.66 - 0.375) = 0.095$$

$$a_2 = 8/24 \int_{0.75}^{1} -\, dt = -8/24\,(1 - 0.75) = -0.0833$$

$$S_1 = \begin{bmatrix} 0.110 \\ 0.095 \\ -0.083 \end{bmatrix}$$

Para S_2 :

$$a_0 = 8/24 \int_{0.375}^{1} 0\, dt = 0$$

$$a_1 = 8/24 \int_{0.66}^{0.75} -1\,dt = -8/24\,(0.66 - 0.375)$$

$$a_1 = -0.095$$

$$a_2 = 8/24 \int_{0.75}^{1} -dt = -8/24\,(1 - 0.66) = -0.1133$$

$$S_2 = \begin{bmatrix} 0.000 \\ -0.095 \\ -0.113 \end{bmatrix}$$

Para S_3 :

$$a_0 = 8/24 \int_{0}^{8/24} dt = 8/24\,(0.33) = 0.11$$

$$a_1 = 8/24 \int_{0.375}^{0.66} dt = 8/24\,(0.66 - 0.33) = 0.11$$

$$a_2 = 8/24 \int_{0.66}^{0.75} dt + 8/24 \int_{0.75}^{1} 3\,dt$$

$$a_2 = 8/24\,(0.75 - 0.66) + 8/24\,(1 - 0.75)$$

$$a_2 = 0.113$$

$$S_3 = \begin{bmatrix} 0.11 \\ 0.11 \\ 0.11 \end{bmatrix}$$

Para S_4 :

$$a_0 = 8/24 \int_0^{0.33} dt = 8/24\,(0.33) = 0.11$$

$$a_1 = 8/24 \int_{0.33}^{0.375} dt + 8/24 \int_{0.375}^{0.66} -2\,dt$$

$$a_1 = 8/24\,(0.375 - 0.33) - 16/24\,(0.66 - 0.375)$$

$$a1 = -0.175$$

$$a_2 = 8/24 \int_{0.66}^{1} -2\,dt = -16/24\,(1 - 0.66)$$

$$a_2 = -0.175$$

$$S_4 = \begin{bmatrix} 0.1100 \\ -0.1750 \\ -0.2266 \end{bmatrix}$$

La representación en el espacio vectorial se muestra en la figura II.17

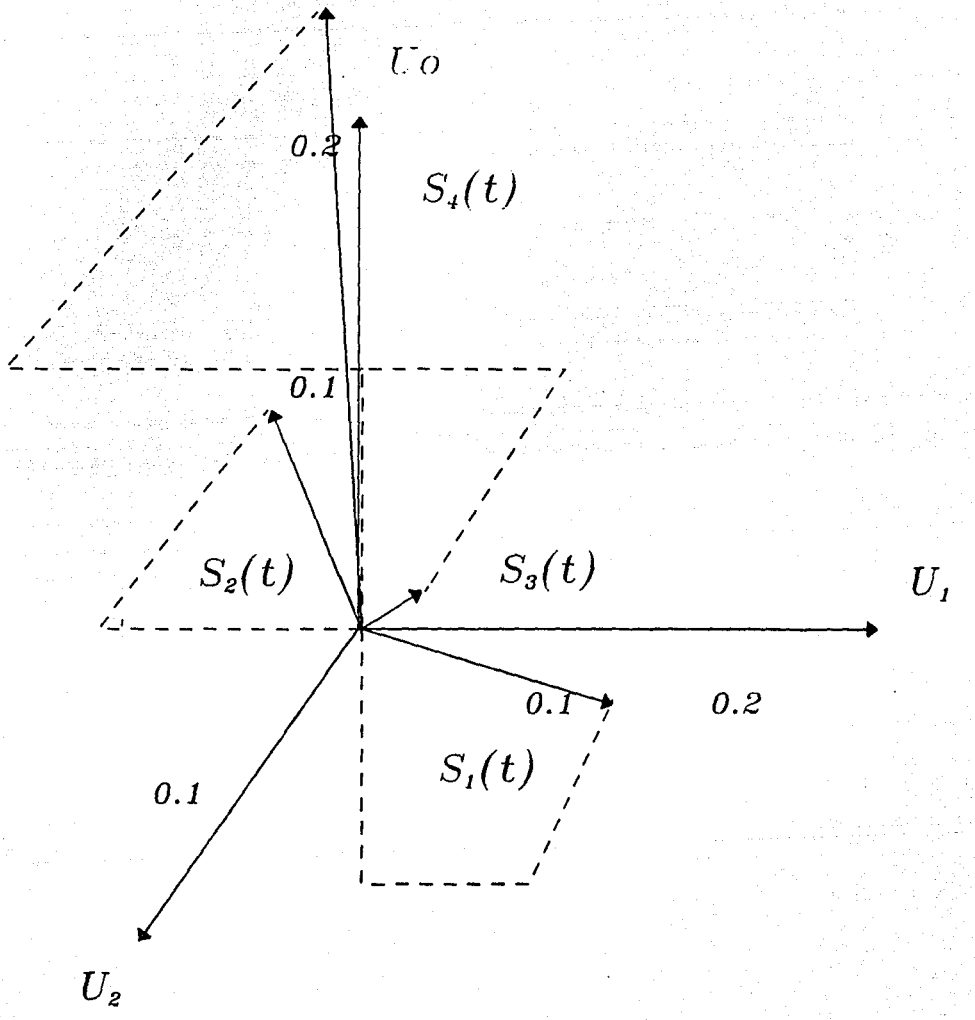


Figura II.17
Representación vectorial para Sn en Un

Esta representación en el espacio vectorial generada por las funciones de Rademacher, es:

$$\{U_n(t)\} = \{\text{rad}(0,t), \text{rad}(1,t), \text{rad}(2,t)\}$$

Para S_1:

$$a_0 = \int_0^{0.375} -dt + \int_{0.375}^{0.75} dt + \int_{0.75}^{1} -dt$$

$$a_0 = -(0.375) + (0.75 - 0.375) - (1 - 0.75)$$

$$a_0 = -0.25$$

$$a_1 = \int_0^{0.375} -dt + \int_{0.375}^{0.5} dt - \int_{0.5}^{0.75} dt - \int_{0.75}^{1} -dt$$

$$a_1 = -(0.375) + (0.5 - 0.375) - (0.75 - 0.5) + (1 - 0.75)$$

$$a_1 = -0.25$$

$$a_2 = \int_0^{0.25} -dt - \int_{0.25}^{0.375} -dt - \int_{0.375}^{0.5} dt + \int_{0.5}^{0.75} dt - \int_{0.75}^{1} -dt$$

$$a_2 = 0.25$$

$$S_1 = \begin{bmatrix} -0.25 \\ -0.25 \\ 0.25 \end{bmatrix}$$

Para S_2:

$$a_0 = \int_{0.375}^{1} -dt = -0.625$$

$$a_1 = \int_{0.375}^{0.5} -dt - \int_{0.5}^{1} -dt = 0.375$$

$$a_2 = -\int_{0.375}^{0.5} -dt + \int_{0.5}^{0.75} -dt - \int_{0.75}^{1} -dt$$

$$a_2 = 0.125$$

$$S_2 = \begin{bmatrix} -0.625 \\ 0.375 \\ 0.125 \end{bmatrix}$$

Para S_3:

$$a_0 = \int_{0.375}^{0.75} dt + \int_{0.75}^{1} 3\,dt$$

$$a_0 = 0.75 + 3\,(1 - 0.75)$$

$$a_0 = 1.5$$

$$a_1 = \int_{0}^{0.5} dt - \int_{0.5}^{0.75} dt - \int_{0.75}^{1} 3\,dt$$

$$a_1 = 0.5 - (0.75 - 0.5) - 3\,(1 - 0.75)$$

$$a_1 = -0.5$$

$$a_2 = \int_0^{0.25} dt - \int_{0.25}^{0.5} dt + \int_{0.5}^{0.75} dt - \int_{0.75}^{1} 3\,dt$$

$$a_2 = 0.25 - (0.5 - 0.25) + (0.75 - 0.5) - (1 - 0.75)$$

$$a_2 = 0$$

$$S_3 = \begin{bmatrix} 1.5 \\ -0.5 \\ 0.0 \end{bmatrix}$$

Para S_4:

$$a_o = \int_0^{0.375} dt + \int_{0.375}^{1} -2\,dt = (0.375) - 2(1 - 0.375)$$

$$a_o = -0.875$$

$$a_1 = \int_0^{0.375} dt + \int_{0.375}^{0.5} -2\,dt - \int_{0.5}^{1} -2\,dt$$

$$a_1 = 0.375 - 2(0.5 - 0.375) + 2(1 - 0.5)$$

$$a_1 = 0.625$$

$$a_2 = \int_0^{0.25} dt - \int_{0.25}^{0.375} dt - \int_{0.375}^{0.5} -2\,dt + \int_{0.5}^{0.75} -2\,dt - \int_{0.75}^{1} -2\,dt$$

$$a_2 = 0.25 - (0.375 - 0.25) + 2(0.5 - 0.375)$$

$$- 2(0.75 - 0/5) + (1 - 0.75)$$

$$a_2 = 0.25$$

$$S_4 = \begin{bmatrix} -0.875 \\ 0.625 \\ 0.250 \end{bmatrix}$$

La representación de este espacio vectorial se muestra en figura II.18:

II.1.10 EL PROCEDIMIENTO DE GRAM-SCHMIDT

En los ejemplos anteriores se observó como es posible representar un conjunto de señales $\{S_n(t)\}$ en un espacio vectorial generado por un conjunto $\{U_n(t)\}$ de señales ortogonales.

Cada vez que se cambia de base $\{U_n(t)\}$, se obtiene una representación diferente de $\{S_n(t)\}$. Se puede preguntar cuál de todas las representaciones es la más adecuada. Como se vio en la ecuación (II.9), el error que puede existir entre la señal $S_m(t)$ real y aquélla que ha sido modelada $S'_m(t)$ depende de la forma de onda de $\{U_n(t)\}$ y de la dimensión del espacio vectorial n. En los casos anteriores siempre se seleccionó n=0, 1, 2. Sin embargo, este número n en teoría puede ser tan grande como se quiera.

Así, al buscar los coeficientes de la serie de Fourier a_o, a_n, b_n, en el ejemplo se podían haber utilizado el número de armónicas que quisieran en lugar de trabajar sólo con la frecuencia fundamental.

De cualquier manera, siempre se puede preguntar cuál es el número mínimo de dimensiones requeridas en la representación vectorial y cuál es el mejor conjunto de funciones base $\{U_n(t)\}$ que se puede utilizar.

Si las señales $\{S_n(t)\}$ fueran mensajes que se quisieran transmitir, recuérdese que al aumentar el número de dimensiones necesarias para representar la señal también aumenta el ancho de banda de la transmisión.

Una manera eficiente para representar un conjunto de M señales, $S_i(t)$, $i = 1, 2, \ldots M$ es aquélla en la que se requieren N dimensiones, donde se debe verificar $N \leq M$.

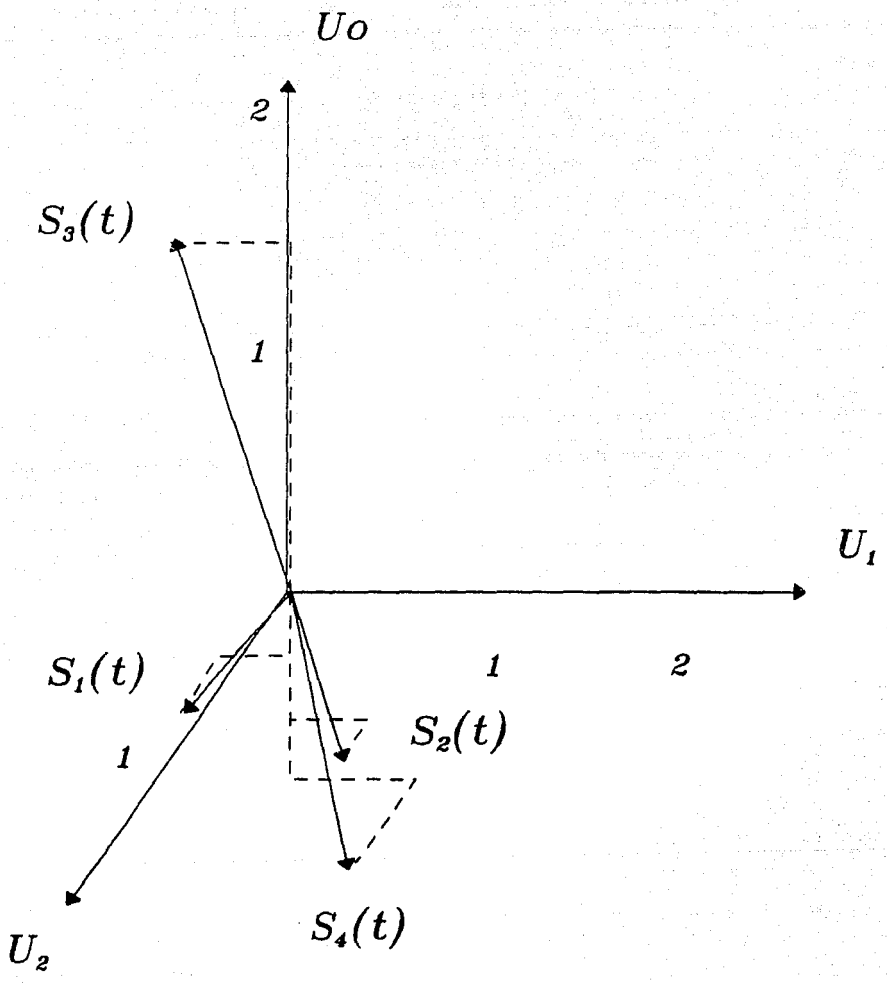


Figura II.18
Representación gráfica de Sn si Un=rad(n,t)

El procedimiento de Gram-Schmidt es un método para encontrar $\{U_N(t)\}$ y a la vez que asegura que N≤M. Se requerirán N=M señales base para generar el espacio vectorial únicamente si las señales $S_i(t)$ son linealmente independientes [es decir, ninguna $S_i(t)$ puede ser expresada como una combinación lineal de las restantes $S_i(t)$]. Si las señales $S_i(t)$ no son linealmente independientes, entonces N < M.

El número de dimensiones, N, requerido para representar un conjunto de señales $\{S_M(t)\} = \{S_1(t), S_2(t), \ldots, S_M(t)\}$ y el correspondiente conjunto de señales base $\{U_n(t)\}$ se puede obtener con el uso del procedimiento de Gram-Schmidt.

Este procedimiento consiste en tomar cualquiera de las señales, obtener su energía y entonces normalizar la señal. El resultado es una señal de forma de onda con energía unitaria.

Esta señal normalizada se toma como la primera función base $U_1(t)$, la cual define la primera dimensión. La representación para otras señales se consigue al encontrar la componente que es obtenida al proyectarlas a lo largo de la primera dimensión; cualquier término que reste es utilizado para definir la señal base para una segunda dimensión. Este proceso se repite hasta que la última de las M señales haya sido descompuesta en las componentes linealmente independientes.

A continuación se da una descripción detallada del procedimiento de Gram-Schmidt:

1. Designar cada una de las funciones como un conjunto

$\{S_n(t)\} = \{S_1(t), S_2(t), \ldots, S_M\}$

2. Encontrar la primera función ortonormal

$$\varphi_1(t) = S_1(t) \;/\; (Es_1)^{1/2}$$

$$\text{donde } Es_1 = \int_0^{To} S_1^2(t)\, dt$$

El vector correspondiente para la forma de onda $S_1(t)$ es

$$S_1 = (S_{11}, 0, 0, \ldots, 0) \qquad \text{donde } S_{11} = (Es_1)^{1/2}$$

3. Encontrar $U_2(t)$

$$\text{Sea } \varphi_2(t) = S_2(t) - S_{21}\,\varphi_1(t) \quad \text{usando}$$

$$S_{ij} = \int_0^{To} S_i(t)\,\varphi_j(t)\, dt$$

$$\text{Si } \varphi_2(t) \le 0, \text{ entonces} \qquad U_2(t) = \varphi_2(t) / (E\varphi_2)^{1/2}$$

donde $E\varphi_j = \int_0^{To} \varphi_j^2 \, dt$

entonces $S_2 = (S_{21}, S_{22}, 0, 0, \ldots, 0)$

donde $S_{22} = (E\varphi_2)^{1/2}$

Si $\varphi_2(t) = 0$, entonces se obtiene

$S_2 = (S_{21}, 0, 0, \ldots, 0)$

otro $\varphi_2(t)$ se encuentra repitiendo este procedimiento donde:

$\varphi_2(t) = S_3(t) - S_{31}\,\varphi_1(t)$

3. Encontrar $\varphi_3(t)$, seleccionar la siguiente forma de onda $S_k(t)$. Evaluar:

$\varphi_3(t) = S_k(t) - S_{k1}\,\varphi_1(t) - S_{k2}\,\varphi_2(t)$

donde $S_{kj} = \int_0^{To} S_k(t)\,\varphi_j(t)\,dt$

Si $\varphi_3(t) = 0$, evaluar $\varphi_3(t)$ hasta $k+1$

Este procedimiento continúa hasta que se utiliza la última señal $S_n(t)$.

Ya que algún $\varphi(t)$ puede ser cero, se tiene entonces $N \leq M$.

Observando un ejemplo:

Sea un conjunto, Sn, de cuatro señales como los que se muestran en la figura II.19

La primera función ortogonal no normalizada es $\varphi 1(t) = S1(t)$ y tiene por energía

$$E\varphi 1 = \int_0^3 S_1^2(t)dt = 3$$

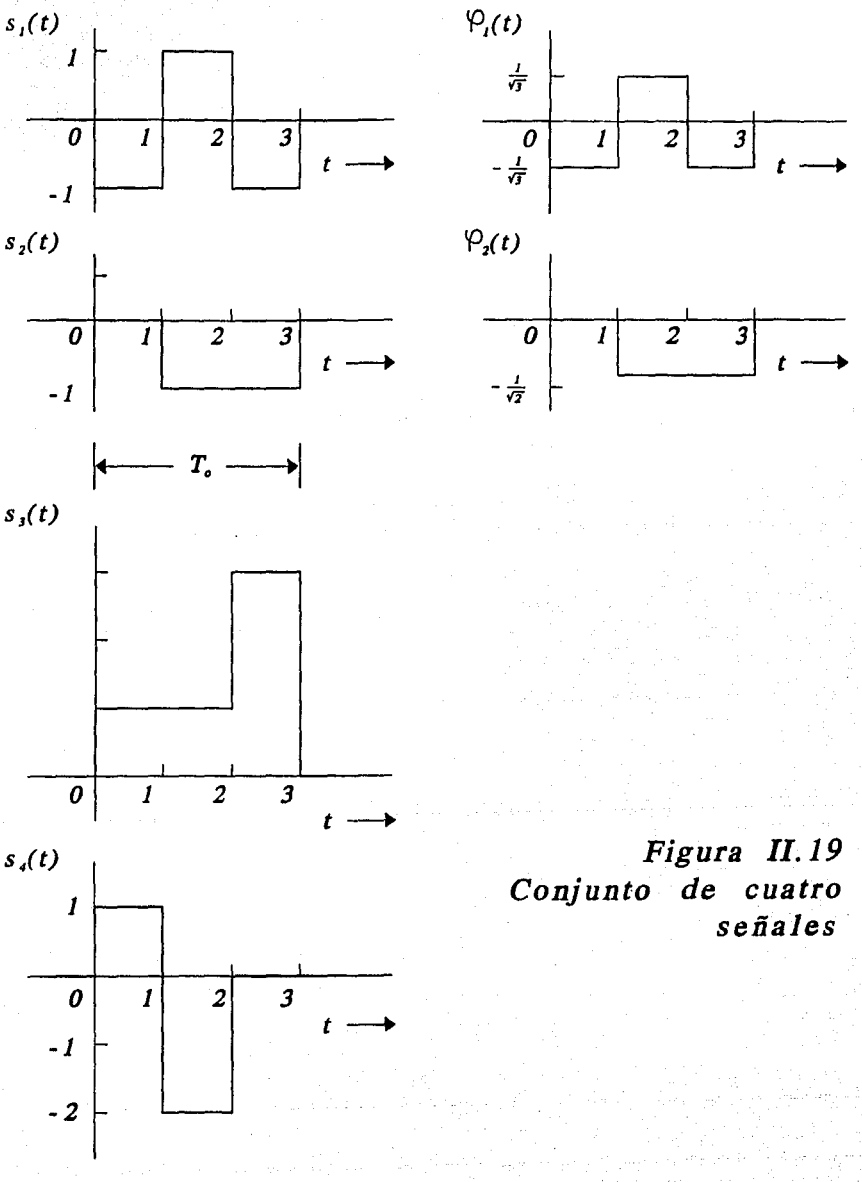


Figura II.19
Conjunto de cuatro señales

Entonces la primera función ortonormal es

$$U1(t) = \frac{S1(t)}{\sqrt{E\varphi 1}} = \frac{S1(t)}{\sqrt{3}}$$

El vector para S1(t) es :

$$S1=(1.73,0,0,0)$$

donde s11 = = 1.73

Evaluando U2(t):

$$S21 = \int_0^3 S2(t)U1(t)dt$$

$$S21 = (0)\left(\frac{-1}{\sqrt{3}}\right) + (-1)\left(\frac{1}{\sqrt{3}}\right) + (-1)\left(\frac{-1}{\sqrt{3}}\right) = 0$$

Entonces:

$$\varphi 2(t) = S2(t) - S21U1(t) = S2(t)$$

$$U2(t) = \frac{\varphi 2(t)}{\sqrt{E\varphi 2}} = \frac{S2(t)}{\sqrt{2}}$$

El vector que determina S2(t) es: S2 = (0,1.41, 0 , 0)

Evaluando el vector U3(t):

$$S31 = \int_0^3 S3(t)\varphi 1dt$$

$$S31 = (1)\left(\frac{-1}{\sqrt{3}}\right) + (1)\left(\frac{1}{\sqrt{3}}\right) + (3)\left(\frac{-1}{\sqrt{3}}\right) = -\sqrt{3}$$

$$S32 = \int_0^3 S3(t)\varphi 2dt$$

$$S32 = (1)(0) + (1)\left(\frac{-1}{\sqrt{2}}\right) + (3)\left(\frac{-1}{\sqrt{2}}\right) = -\frac{4}{\sqrt{2}}$$

Entonces:

$$\varphi 3(t) = S3(t) - S31U1(t) - S32U2(t)$$

$$\varphi 3(t) = S3(t) - (-\sqrt{3})U1(t) - \left(-\frac{4}{\sqrt{2}}\right)U2(t) = 0$$

como $\varphi 3(t)$ es cero no se necesita de esta señal para describir S3(t). El vector de señal es:

$$S3 = (-1.73, -2.83, 0)$$

Se sigue buscando $\varphi 3(t)$

$$\varphi 3(t) = S4(t) - S41U1(t) - S42U2(t)$$

En donde:

$$S41 = \int_0^3 S4(t)\varphi 1 dt$$

$$S41 = (1)\left(\frac{-1}{\sqrt{3}}\right) + (-2)\left(\frac{1}{\sqrt{3}}\right) + (0)\left(\frac{-1}{\sqrt{3}}\right) = -\sqrt{3}$$

$$S42 = \int_0^3 S4(t)\varphi 2 dt$$

$$S42 = (1)(0) + (-2)\left(\frac{-1}{\sqrt{2}}\right) + (0)\left(\frac{-1}{\sqrt{2}}\right) = \sqrt{2}$$

Entonces:

$$\varphi 3(t) = S4(t) - S41U1(t) - S42U2(t)$$

$$\varphi 3(t) = S4(t) - (-\sqrt{3})U1(t) - \sqrt{2}U2(t) = 0$$

Como $\varphi 3(t)$ es cero el vector S4 queda definido por:

$$S4 = (-1.73, 1.41)$$

Una representación gráfica de las señales se muestra a continuación en la figura II.20.

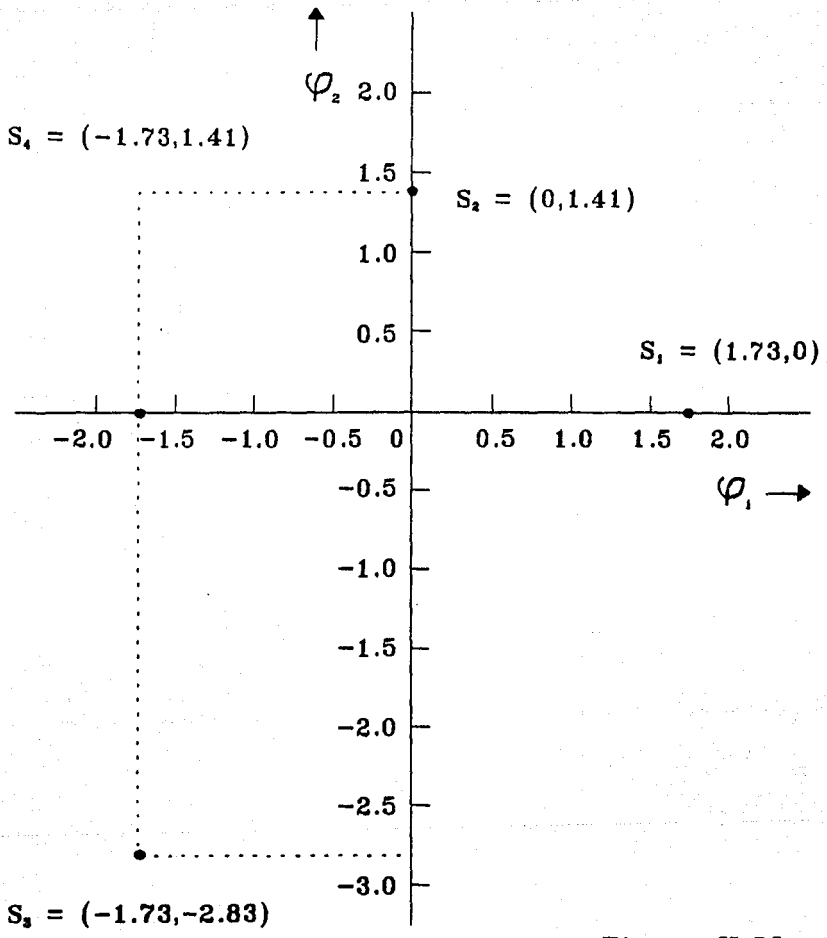


Figura II.20
Representación gráfica del ejemplo de Gram–Schmidt

III EL PROCEDIMIENTO DE GRAM-SCHMIDT APLICADO AL RECONOCIMIENTO DE VOZ POR COMPUTADORA

III.1 EL TMS320C30 Y SUS HERRAMIENTAS DE DESARROLLO

El procesamiento digital de señales conlleva una gran cantidad de aplicaciones. Algunos ejemplos incluyen filtros digitales, reconocimiento de voz, procesamiento de imágenes y audio digital. Estas aplicaciones tienen como características en común:

* Algoritmos con cálculos matemáticos intensos
* Operación en tiempo-real
* Muestreo de funciones continuas
* Flexibilidad del sistema

Para el caso del reconocimiento de voz en el procedimiento de Gram-Schmidt se debe calcular:

$$S_{1\,1} = \int s(t)\ U_1(t)$$

de manera discreta

$$S_{1\,1} = \sum s(t)\ U_1(t)$$

si s(t) y $U_1(t)$ contienen 5000 muestras cada uno, aproximadamente se deben realizar 25,000,000 de multiplicaciones lo que significa un trabajo matemático intenso.

Además de esto, el algoritmo se debe realizar en tiempo real. En reconocimiento de voz, un retardo entre la pronunciación de una palabra y su reconocimiento, puede ser inaceptable. El procesador debe ser capaz de mantener una gran cantidad de información muestreada y también realizar cálculos aritméticos en tiempo real.

Flexibilidad del sistema

El reconocimiento de voz en el procedimiento de Gram-Schmidt, es de momento, una técnica inexacta que requiere modificar continuamente el programa. Por ello es necesario tener un procesador programable que otorgue esta flexibilidad.

El avance que han tenido los circuitos integrados se ha visto reflejado en aplicaciones de DSP que se han extendido a las telecomunicaciones tradicionales, a procesadores de gráficas e imágenes y luego al procesamiento de audio.

Un desarrollo avanzado en tecnología DSP es la familia de procesadores TMS320. Esta familia incluye tres generaciones de procesadores. La primera familia de estos procesadores (TMS32010) daba al diseñador la posibilidad de realizar 5 millones de operaciones en punto flotante por segundo. La segunda generación incluye los TMS32020 y TMS32023. El TMS32025 puede realizar 10 millones de operaciones en punto flotante por segundo. Posteriormente se añadió un espacio de memoria expandida, se realizó un ciclo especial de multiplicación y funciones expandidas de I/O que dieron a los TMS32020 de dos a cuatro veces una eficiencia mayor que la de sus predecesores. La tercera generación de la familia TMS320, los TMS32030, tienen una velocidad de proceso de 33 millones de operaciones de punto flotante por segundo (MFLOPS).

Arquitectura básica del TMS320

El TMS320 es un dispositivo que permite cumplir con los objetivos trazados, referentes a realizar cálculos matemáticos intensos y trabajar con una gran rapidez.

Lo anterior se consigue ya que el TMS320 cuenta con las siguientes características:

* Arquitectura Harvard
* "Pipeline" extensivo
* Multiplicador dedicado por hardware
* Instrucciones especiales de DSP
* Ciclo de máquina de 60 ns

Además los dispositivos TMS320 son programables permitiendo la flexibilidad y fácil uso del sistema.

Arquitectura Harvard

Este tipo de arquitectura permite una transferencia de información entre el programa y el espacio de datos. Además se mantienen separados el bus de datos y el del programa permitiendo una ejecución a máxima velocidad. De esta manera se aumenta el poder de procesamiento.

"Pipeline"

El "Pipeline" se utiliza para reducir el ciclo de tiempo de instrucción a un mínimo. En el TMS320 se pueden realizar de dos a cuatro instrucciones en paralelo. Las instrucciones: "fetch", "decode" y "execute" se pueden realizar en diferentes niveles al mismo tiempo; Durante un ciclo de máquina es posible que la n'ésima instrucción sea buscada mientras la instrucción n-1 es decodificada y la instrucción n-3 es ejecutada.

Multiplicador dedicado por hardware

Como se pudo observar anteriormente, el procedimiento de Gram-Schmidt utiliza una gran cantidad de sumas y multiplicaciones; por lo tanto, mientras más rápido sea el procesador para ejecutarlas, más eficiente será el sistema.

En los procesadores de uso general, una multiplicación se realiza mediante una serie de sumas utilizando muchos ciclos de máquina. En los procesadores de DSP se tiene un "multiplicador dedicado". En un TMS320 la multiplicación utiliza un ciclo de máquina.

De manera general, realizar una multiplicación requeriría de 30 a 40 ciclos de máquina en un procesador de uso general, mientras que un TMS320 ocuparía las instrucciones:

LT:	cargar multiplicando en el registro T
DMOV:	llevar el dato a la memoria
MPY:	multiplicar
APAC:	sumar el resultado de la multiplicación al acumulador

es decir, cuatro ciclos de máquina.

Funciones especiales de DSP

Como se observó en el ejemplo previo, existen funciones especiales como LT, DMOV, MPY, APAC que tienden a reducir el número de ciclos de máquina en cada operación. Existe otra función especial en el TMS320, ésta es: RPTR, "repeat the next instruction N times" y MACD: "Make LT, DMOV, MPY and APAC".

Ciclo de máquina

Gracias a las características que se han mencionado previamente, combinado con el diseño del circuito integrado se tiene un DSP con una gran velocidad de proceso. Cada ciclo de máquina en un TMS320C30, por ejemplo, es de 60 ns.

El TMS320C30

El TMS320C30 pertenece a la tercera generación de procesadores TMS320, posee una velocidad computacional de 33 MFLOPS (millones de operaciones punto flotante por segundo). Con este procesador, el usuario tiene el apoyo de herramientas de software dándole mayor eficiencia al sistema.

Las principales características del TMS320C30 son:

* Ciclo de instrucción de 60 ns
* (2) memorias RAM 1 K * 32 bits
* (1) memoria ROM 4 K * 32 bits
* Cache de instrucción 64 x 32 bits
* Palabras de 32 bits de instrucción y de datos
* Multiplicador dedicado de 32 bits punto flotante
* ALU para números punto flotante 32 bits
* Controlador de acceso directo a memoria (DMA) en el chip
* Soporte con lenguaje de alto nivel

El CPU del TMS320C30

El CPU consiste de los siguientes elementos: multiplicador de punto flotante, ALU para realizar operaciones en punto flotante y operaciones lógicas, registros aritméticos auxiliares y los buses asociados. El TMS320C30 tiene la capacidad de realizar, en un sólo ciclo, multiplicaciones y sumas de manera paralela. Gracias a ésto, el TMS320C30 alcanza su máxima capacidad computacional de 33 MFLOPS.

El CPU contiene 28 registros, los cuales pueden ser operados por el multiplicador o el ALU. Los ocho primeros de estos registros (R0-R7) son registros de precisión que soportan operaciones de números de hasta 40 bits en punto flotante. Los siguientes ocho registros (AR0-AR7) tienen por función primaria la generación de direcciones de memoria. Sin embargo, pueden ser usados como registros generales de 32 bits. Dos registros auxiliares ARAU0 y ARAU1 pueden generar dos direcciones en un sólo ciclo de instrucción. Estos registros pueden trabajar en paralelo con la ALU y el multiplicador. Además pueden realizar corrimiento en las direcciones y funcionar como registros de índice.

Los registros restantes soportan una gran variedad de funciones: direccionamiento, manejo del stack, interrupciones.

Organización de datos

El espacio total de memoria en el TMS320C30 es 10 M (millones) x 32 bits. Una palabra del TMS320 es de 32 bits y todos los direccionamientos son por palabras. Programa, datos y espacio para I/O está contenida en los 16 M-palabra de memoria. Los bloques de RAM 0 y 1 son cada uno de 1 K x 32 bits. El bloque de ROM es de 4 K x 32 bits.

Cada bloque de memoria RAM y ROM pueden soportar un acceso en un ciclo de instrucción; es decir, el usuario puede en un sólo ciclo de instrucción obtener una palabra de datos de RAM y una palabra de programa de ROM. Los diferentes buses de programa y datos permiten hacer lecturas y escrituras en paralelo.

"EL DMA"

El controlador DMA puede realizar lecturas y escrituras hacia cualquier localidad en la memoria sin tener que interferir con las operaciones del CPU. De esta manera se pueden interfasar memorias y periféricos LA/D, puertos seriales, etc., sin afectar el trabajo del CPU.

El controlador de DMA contiene sus propios generadores de direcciones y contadores. El DMA responde a las interrupciones de manera similar al CPU. Esta capacidad le permite al DMA transferir datos basados en las interrupciones recibidas. La transferencia de datos puede ser realizada por el DMA en lugar del CPU sin tener conflicto con este último.

"Pipelining" en el TMS320

La operación en el TMS 320C30 está controlada por cinco unidades principales:

Fetch Unit (F). La cual controla y actualiza los contadores del programa y trae una palabra de memoria.

Decode Unit (D). Quien decodifica una palabra de instrucción y genera una dirección.

Read Unit (R). Controla las lecturas de memoria.

Execute Unit (E). Realiza operaciones y escrituras de regreso a memoria.

DMA Channel (DMA). Tiene la posibilidad de leer y escribir en memoria paralelamente a la operación del CPU.

Cada instrucción posee cuatro de estas etapas: Fetch, decode, read y execute. El empalmar estas etapas se conoce como "Pipelining".

Herramientas de desarrollo y soporte

Herramientas de Software:

Assembler/Linker: el macro ensamblador traduce el lenguaje ensamblador en código ejecutable. El linker permite enlazar programas que han sido desarrollados por separado.

C Compiler: El compilador en C es una versión completa del lenguaje C definido por Kernighan y Ritchie.

Spox: Este es un software de interfaz entre el usuario y la tarjeta "banshee board" conteniendo el TMS320C30. Al momento de instalar Spox en la computadora, se inicializa y verifica el buen funcionamiento de la tarjeta.

Las utilidades de Spox son:

- Permite la utilización de un editor para programar el TMS320C30 con la estructura del lenguaje C

- Posee funciones predefinidas necesarias para configurar el TMS320C30 como son

*Reservar memoria	SA_create
*Liberar memoria	SA_free
*Abrir o cerrar canal de grabación	SIG_open
*etc.	

- Posee diferentes Bibliotecas con funciones propias de DSP y operaciones con vectores y matrices

*Transformada de Fourier	SV_Ft
*Multiplicación de vectores	SV_mul
*Resta de vectores	SV_dif
*Multiplicación de matrices	SM_mul
*Hacer producto punto	SV_dotp
*Crear un vector	SV_create
*Llenar un vector	SV_fill
*Copiar un vector	SV_assaign
*etc.	

Como se puede apreciar el sistema de procesamiento puede adquirir una gran eficiencia al trabajar con un TMS320C30. Sin embargo otras limitaciones como por ejemplo la capacidad de memoria hacen que aplicaciones como el reconocimiento de voz sea aún un proyecto a mejorar.

III.2 PROGRAMA DE RECONOCIMIENTO EN EL TMS320C30

En la sección anterior se observó el caso de un conjunto de señales $\{S_n(t)\}$, pulsos rectangulares, con los que se formó un conjunto de señales rectangulares $\{U_n(t)\}$. El error entre la señal modelada $S_n'(t)$ y la señal verdadera $S_n(t)$, e = E($S_n(i)$ - S*n(i)) es prácticamente nulo.

Es posible pensar enconces que si se tiene una nueva señal, f(t), tal que f'(t) = Sn'(t), las componentes de f(t) en {Un} deben ser prácticamente las mismas que Sn(t).

En los espacios vectoriales esta característica es aprovechada para realizar reconocimiento de voz. Para lograr el reconocimiento se realizan varias etapas, pues primero es necesario generar el espacio vectorial para después poder realizar el reconocimiento.

La figura III.1 muestra las etapas de las que consta este reconocedor.

La primera fase es la de entrenamiento; en esta parte el locutor graba todas las señales que servirán para formar el espacio vectorial. La segunda fase es la que construye el epacio vectorial que servirá de referencia para el reconocimiento. La última fase es el reconocedor quien, como lo explicamos anteriormente, también utiliza el procedimiento de Gram-Schmidt.

En seguida se verá con más detalle cómo está compuesta cada una de estas partes.

III.2.1 FASE DEL ENTRENAMIENTO

El locutor debe grabar todas las señales que servirán de base para posteriormente generar el espacio vectorial.

El diagrama general del entrenamiento del sistema se muestra en la figura III.2.

Como se puede observar en el diagrama, se hace uso de dos funciones: JL_faq y JL_qb. La primera función se encuentra dentro de una librería que se ha definido como AD1H.H. La función se encarga en sí de cuatro aspectos fundamentales:

1) Inicializar el canal de la tarjeta de adquisición que se piensa utilizar.

En este caso se está utilizando la tarjeta "Madre" "Banshee Board" que posee el TMS320C30, así como la tarjeta "Hija" "AD16". La AD16 posee dos canales de entrada y dos canales de salida para señal analógica. Por esto mismo se debe seleccionar qué canal se va a utilizar.

2) Determinar la velocidad del muestreo.

La velocidad del muestreo no tiene porque ser forzosamente de 8,000 Hz, aunque sea lo más recomendable. El usuario puede cambiar este parametro sin ningún problema.

3) Seleccionar el tipo de datos.

Al momento de estar muestreando, cada vez que se toma un dato es necesario saber qué formato se le va a dar para después poderlo leer correctamente. En este caso se ha seleccionado el formato de flotantes con signo en 32 bits (C30-float), pero también están disponibles por ejemplo: enteros con o sin signo en 16 y 32 bits, o flotantes con o sin signo en 16 y 32 bits.

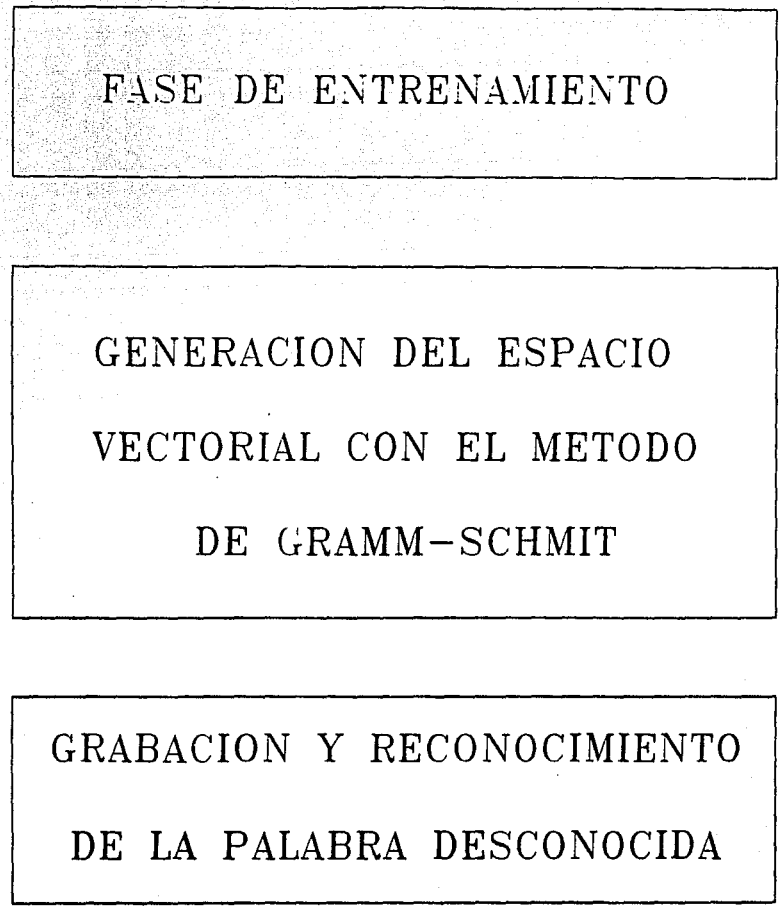
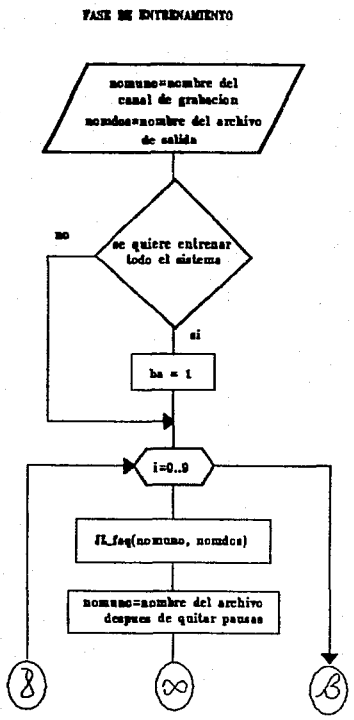


Figura III.1 Forma general del sistema de reconocimiento



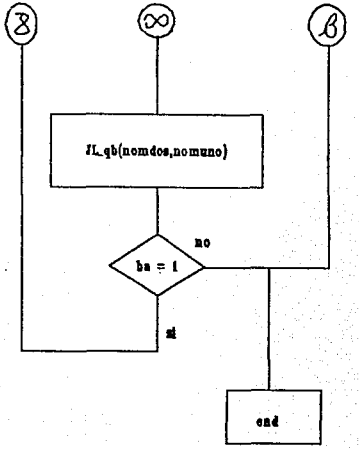


Figura III.2 Diagrama para el entrenamiento para el reconocedor de palabras aisladas

4) Determinar la longitud del archivo de señal que se va a generar.

Como default se tiene un BUFSIZE de 120 en un ciclo iterativo de 60, lo que da un total de 7200 muestras a grabar en 0.9 s, suficiente para decir algún número del cero al nueve.

La figura A.1 del apéndice A muestra el diagrama de JL_faq.

Cuando se registran las 7,200 muestras aproximadamente la mitad de éstas corresponde a un silencio del locutor. Si se trabaja con todas estas muestras, se aumentaría en gran medida el número de cálculos que deben realizarse en sumas y multiplicaciones sin corresponder realmente a una información valiosa.

Debido a ésto, las pausas antes y después de decir la palabra deben de quitarse de la señal a analizar. La función JL_qb está encargada de ésto.

Para lograr la limpieza de la señal, el algoritmo se basa en lo siguiente:

Sea x (t) una señal análogica; si ésta es muestreada por un convertidor analógico-digital de tal manera que puede representarse mediante N muestras, entonces:

$$x(t) = \sum_{i=0}^{N-1} x^*(i)$$

La señal x*(t) se puede representar mediante tres sumatorias:

$$x^*(t) = \sum_{i=0}^{L} x^*(i) + \sum_{L+1}^{M} x^*(i) + \sum_{M+1}^{N-1} x^*(i)$$

Si $L << M-(L+1)$

$$P_1^*(t) = \sum_{i=0}^{L} x^*(i)$$ representa la primera pausa

Si $M-(L+1) >> (N-1)-(M+1)$

$$M^*(t) = \sum_{L+1}^{M} x^*(i)$$ representa el mensaje de la palabra

$$P_2^*(t) = \sum_{M+1}^{N-1} x^*(i)$$ representa la segunda pausa

L y M son dos muestras anteriores a N tal que $L < M < N$

Se busca obtener la energía E_1 para $P_1^*(t)$

$$dotp_1 = \sum_{i=0}^{L} x^*(i)\ x^*(i)$$

$$E_1 = (dotp_1)^{1/2}$$

E_1 será entonces la energía contenida en un marco de L muestras de la primera pausa. Si toda la señal se divide en marcos de longitud L, entonces se tendra que la energía total de la señal será:

$$E_{tot} = \sum_{i=1}^{r} E_i$$

donde r = int (N/L) r es es el entero más grande de (N/L)

$$E_i = \sum_{q=L_i}^{q=(L-1)(i+1)} x^*(q)\ x^*(q)$$

$i = 0, 1, \ldots. r-1$ Si $i < r$

$$E_i = \sum_{q=L(i-1)}^{N-1} x^*(q)\ x^*(q)$$

Si $i = r$

La energía del mensaje estará contendia en algunos miembros del conjunto $\{E_n\}$ $\{E_2, E_3, E_4, \ldots, E_r\}$. Experimentalmente se ha encontrado que $E_1 >> En$ $n = 2,3, \ldots, r$.

La forma de onda se muestra en la figura III.3

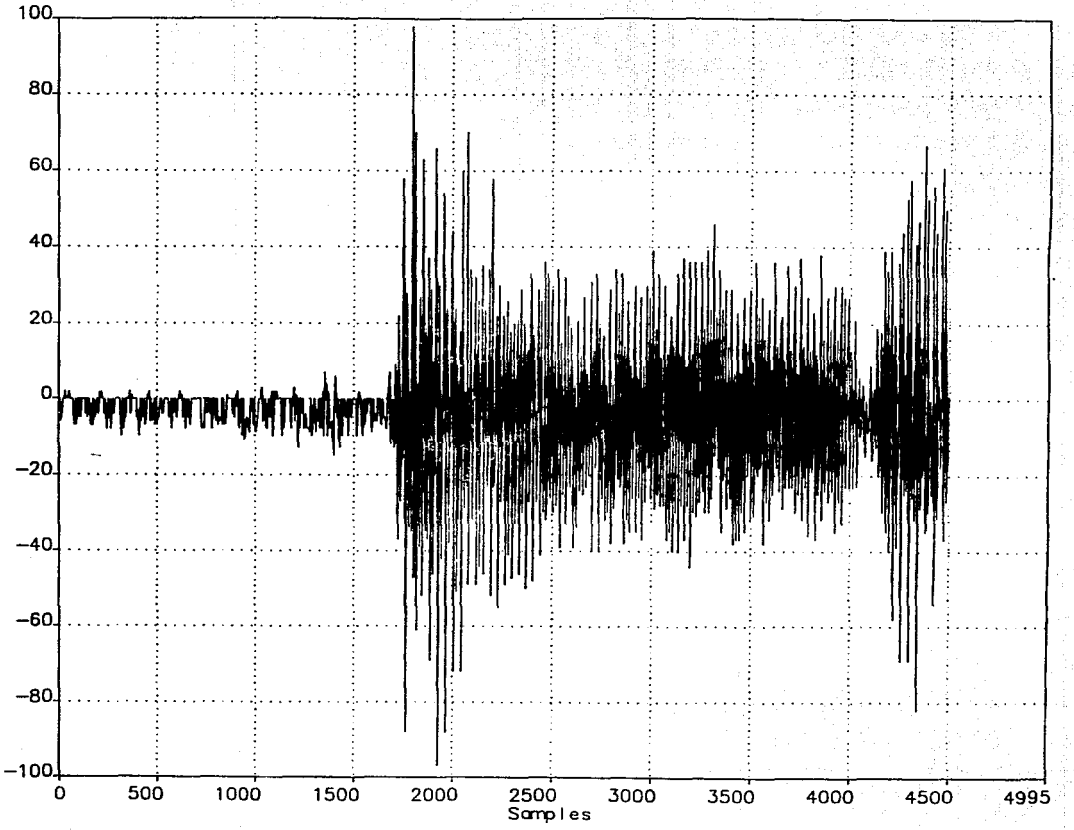


Figura III.3 Forma de onda de un digito

Es posible notar cuando empieza la palabra, ya que la amplitud y la frecuencia de la señal empiezan a aumentar considerablemente. Justamente la gran cantidad de cruces por cero hace que la energía contenida en el mensaje sea menor a la energía contenida por el ruido de la señal.

La función JL_qb hace uso de esta característica y de un ciclo iterativo para limpiar las señales.

Sea V_1^* (t) el primer marco de la señal de un longitud L, V_2^* (t) cualquier marco sucesivo de la señal. dotpr y dotpr2 son las energías asociadas respectivamente a $V_1^*(t)$ y V_2^* (t).

El ciclo iterativo sería de la siguiente manera:

Si la energía de $V_2^*(t)$ es menor que la de $V_1^*(t)$, entonces se trata de una parte del mensaje que hay que guardar. De lo contrario, es una pausa, pero que no se desechará de inmediato. Este marco de muestras es guardado en una matriz.

Si en un momento dado entra otra parte de mensaje ($V_2 < V_1$) con la matriz semillena, entonces quiere decir que las pausas que se habían guardado forman parte de la palabra a reconocer y por lo tanto deben ser restituidas a la señal.

Si la matriz se llena, quiere decir que son pausas a los extremos de la palabra y por tanto no son restituidos a la señal.

La figura III.4 muestra una explicación gráfica del funcionamiento y la figura A.2 en el apéndice A muestra todo el diagrama de flujo de este programa.

III.2.2 FASE DE RECONOCIMIENTO

Una vez que en la parte de entrenamiento se han grabado y limpiado las señales de voz de las pausas iniciales y finales, éstas deben de codificarse para pasar a ser señales patrón que posteriormente servirán al reconocimiento.

Como se indicó en el capítulo de Señales Ortogonales, cada palabra será proyectada sobre los ejes de una base ortogonal. Cada uno de los coeficientes resultado de la proyección, serán los componentes de la palabra y por tanto la codificación de cada una de la señales de voz. La base ortogonal será generada por las mismas señales siguiendo el procedimiento de Gram-Schmidt.

Después que las señales de voz salen de la función JL-qb se tiene un conjunto de señales $\{S_n\}$:

S_0 = cero
S_1 = uno
S_2 = dos
.
.
.
S_9 = nueve

Cada una de estas señales es de forma de onda distinta y con un número diferente de muestras. A partir de ellas, se formará el espacio vectorial.

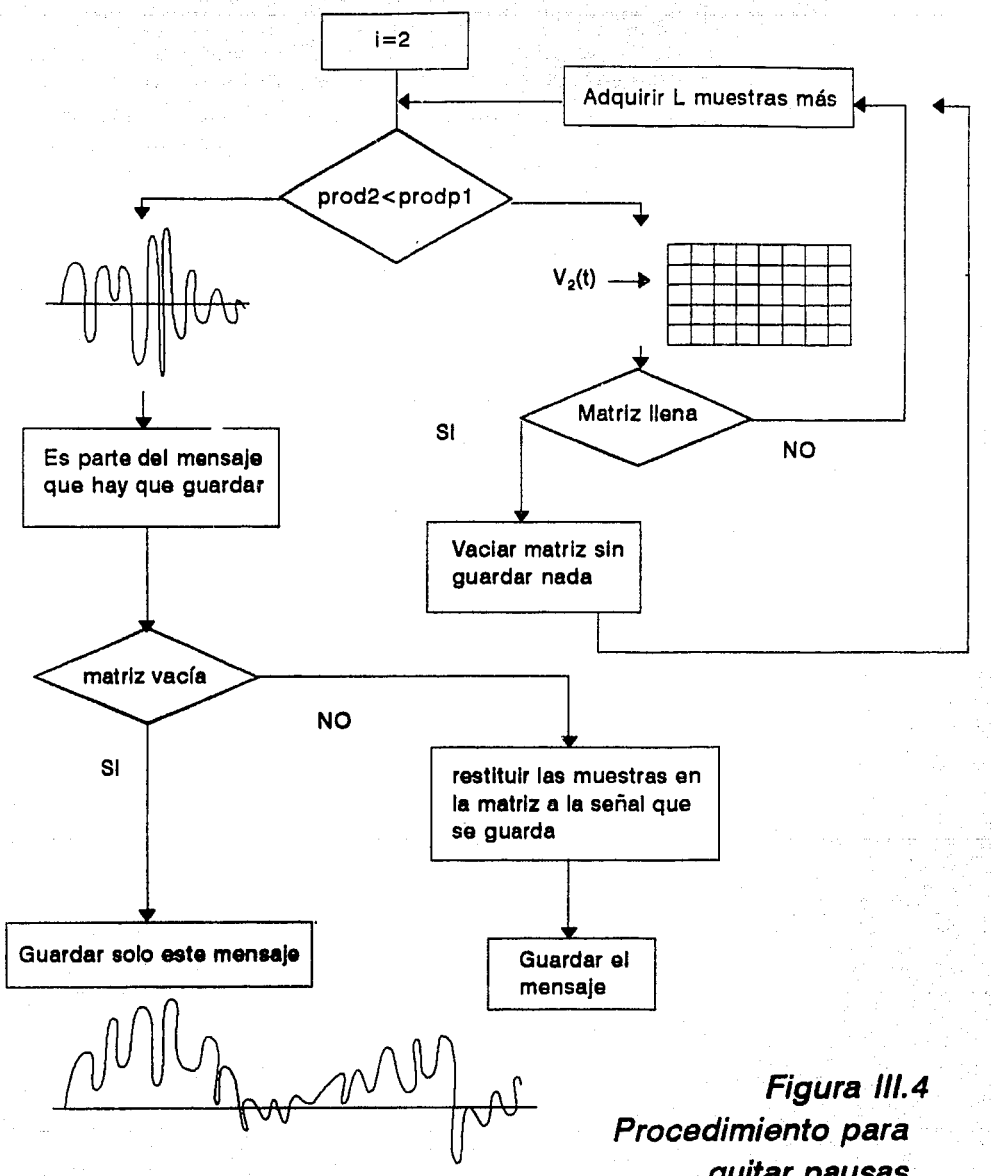


Figura III.4
Procedimiento para quitar pausas

El diagrama de flujo del programa que genera el espacio vectorial, se muestra en la figura A.3 del apéndice A.

Como puede observarse, existen dos funciones que no se han mencionado anteriormente: JL_dotp y JL_subs. El recordar el procedimiento de Gram-Schmidt, se requiere de obtener el producto punto entre dos señales y la resta entre los mismos.

Sin embargo, no existe definición matemática para la multiplicación de dos vectores con diferentes longitudes, mucho menos la resta de un vector que tiene un número mayor de componentes que otro.

El problema anterior se resuelve haciendo uso de la programación dinámica.

III.2.3 PROGRAMACION DINAMICA

La programación dinámica es un concepto matemático utilizado para el análisis de procesos con toma de decisión secuencial.

La programación dinámica fue popularizada por Bellman en los años 50´s. Esta fue propuesta rápidamente para ser utilizada en el reconocimeinto de voz y se aplicó tan pronto como surgieron las computadoras digitales con suficiente memoria; eso fue alrededor de 1962.

Teóricamente la idea de programación dinámica está basada en una simple propiedad de procesos de decisión llamada "el principio del rendimiento óptimo".

Este principio de tiene la propiedad de que cualquiera que sea el estado y la decisión incial, las decisiones restantes deben constituir una estrategia óptima.

Típicamente la programación dinámica aplicada al reconocimiento de voz se ha dirigido a la alineación en tiempo entre dos diferentes segmentos de voz.

III.2.4 UN EJEMPLO PARA INTRODUCIR LA PROGRAMACION DINAMICA.

Consideremos el problema en que se quiere hacer el producto punto entre dos señales.

Sean S_1 y S_2 dos señales como las que se muestran en la figura III.5

Si estas señales se muestrean a razón de cuatro muestras en t_0 segundos, se obtenien los siguientes resultados:

$$S1=\begin{bmatrix}1\\1\\-1\\-2\end{bmatrix} \qquad \text{y}$$

$$S2=\begin{bmatrix}2\\0\\-2\\2\end{bmatrix}$$

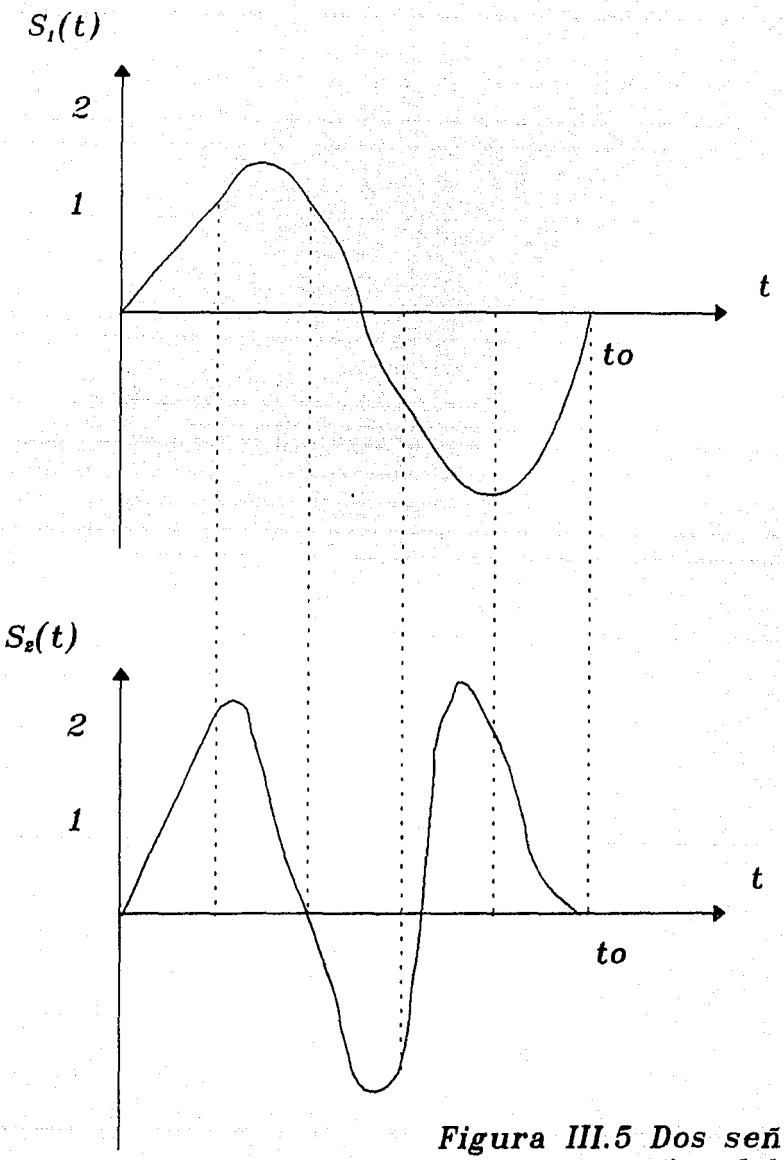


Figura III.5 Dos señales de igual longitud

El producto punto está dado por:

$$prodp = \sum_{i=0}^{N-1} S_1(i) \quad S_2(i)$$

donde N es el número total de muestras.

Una manera gráfica de ilustrar el producto punto sería dibujando una matriz de NxN y los vectores de señal S_1 y S_2 abajo y a la izquierda respectivamente

-2				
-1				
1				
1				
	2	0	-2	2

De acuerdo a la definición que se dio anteriormente del producto punto se buscaría llenar la diagonal principal de la matriz.

Cada casilla de la diagonal de la matriz MAT estará dada por:

$MAT[0][0]=S_1(0) \times S_2(0)$

$MAT[i][j]=S_1(j)xS_2(i) + MAT[i-1][j-1]$

$i=j=1,2,3$

La matriz resultante para esta operación será:

			0
		4	
	2		
2			

Es decir el resultado del producto punto entre estas dos señales es MAT[3][3]=0, S_1 y S_2 son ortogonales entre ellas.

Observemos ahora un caso en que S_1 y S_2 no son de la misma longitud. La figura III.6 muestra estas dos señales.

Como se puede observar, las dos señales son muy parecidas aunque la longitud es diferente.

Nuevamente se buscará el producto punto entre S_1 y S_2. Se observará el procedimiento gráfico sobre una matriz MAT.

Para caracterizar las señales, se muestrean a razón de N muestras por segundo. Se Obtiene:

$$S_1 = \begin{bmatrix} 1 \\ 4 \\ -3 \\ 2 \end{bmatrix}$$

$$S_2 = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 2 \\ -2 \\ -3 \\ -1 \\ 2 \end{bmatrix}$$

ns1 es el número de muestras de S_1 y
ns2 es el número de muestras de S_2

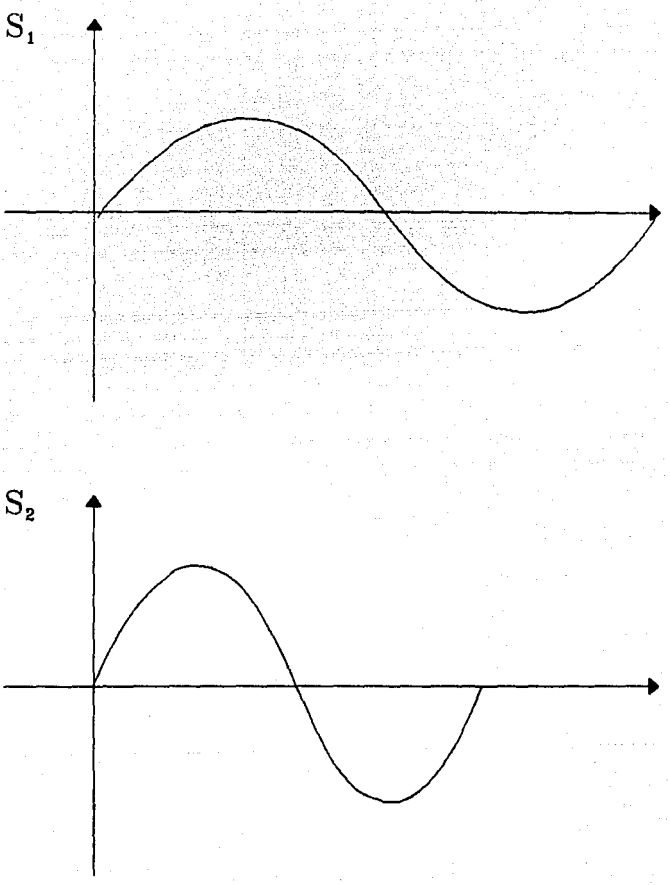


Figura III.6 Dos señales de diferente longitud

2								
-3								
4								
1								
	1	2	3	2	-2	-3	-1	2

Para completar la matriz se usará la programación dinámica. El procedimiento consiste en dos pasos:

1) Cada elemento de la matriz deberá corresponder a:

MAT[0][0]=S1(0)xS2(0)

$$MAT[i][j]= S2(i)xS1(j) + \text{argmax} \begin{bmatrix} MAT[i-1][j] \\ MAT[i-1][j-1] \\ MAT[i][j-1] \end{bmatrix}$$

si i=1.....ns2
j=1.....ns1

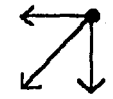
MAT[0][i]=S2(i)xS1(0)+MAT[0][i-1]

i=1.....ns2

MAT[j][0]=S1(j)+S2(0)+MAT[j-1][0]

j=1.....ns1

En realidad lo que se hace es maximizar el producto punto de tal manera que para cada casilla de la matriz se busca el máximo predecesor j . La búsqueda es del siguiente tipo:



Las flechas indican los tres candidatos que pueden añadirse al producto punto.

Si se está en el primer renglón, la búsqueda será:

i-1 ← si j=0

si se está en la primera columna, la búsqueda será:

j-1 si i=0

El resultado al llenar la matriz es el siguiente:

2	5	13	25	32	32	38	47	56
-3	2	7	16	26	38	47	50	46
4	3	25	32	24	12	8	16	22
1	1	3	6	8	6	3	2	4
	1	2	3	2	-2	-3	-1	2

2) Una búsqueda hacia atrás de los máximos elementos dá la ruta que se ha seguido para obtener el producto punto. La flecha en la figura anterior muestra este camino.

En este caso el resultado del producto punto es de 56.

Como se observa, al introducir la programación dinámica el cálculo del producto punto se convierte en una función no lineal.

III.2.5 RESTRICCIONES A LA PROGRAMACION DINAMICA

En realidad existen diferentes tipos de decisión para encontrar el camino optimo que la flecha muestra en la figura anterior.

En la figura III.7 se pueden observar los algoritmos de búsqueda para A) Itakura, B) el algoritmo de Itakura modificado, y C) el de Sakoe-Chiba.

El algoritmo que se utiliza en este programa es el de Sakoe-Chiba. Puede apreciarse en este algoritmo como está permitida la búsqueda de manera vertical.

Si se observa la matriz MAT, cada casilla MAT[i][j] que es atravesada por la flecha significa que $S_1[j]$ y $S_2[i]$ son muy parecidos. De esta manera en este ejemplo se tiene:

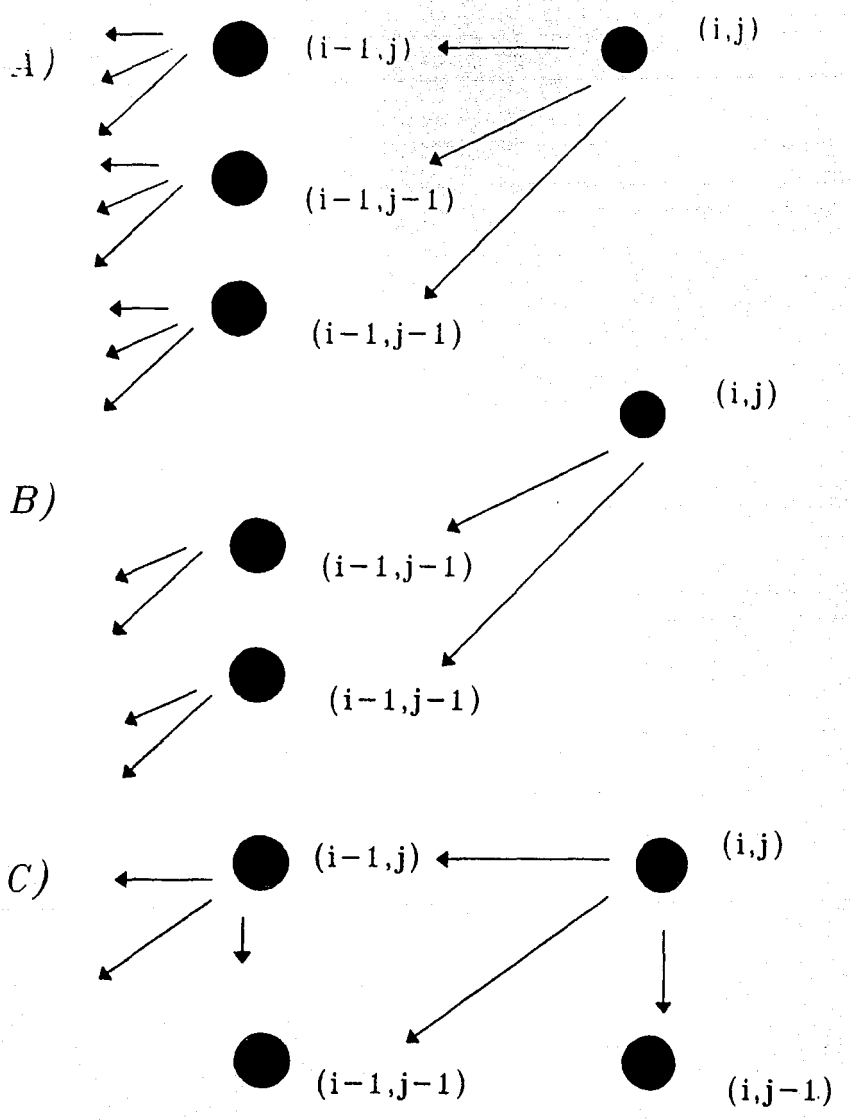


Figura III.7 Diferentes algoritmos de busqueda

S2(0)	se parece a	S1(0) y S1(1)
S2(1)		S1(1)
S2(2)		S1(1)
S2(3)		S1(1)
S2(4)		S1(1)
S2(5)		S1(1)
S2(6)		S1(2)
S2(7)		S1(3)

Se puede afirmar si se toma como base la señal S_1;

El elemento 0 de S_2 sufrió una expansión para parecerse a 0,1 de S_1.

Los elementos 0,1,2,3 de S_2 sufrieron compresión para parecerse a 1 elemento 1 de S_1.

Los elementos 4,5,6 de S_2 sufrieron compresión para parecerse al elemento 2 de S_1.

El algoritmo propuesto por Itakura no permitiría la expansión de la señal S_2 y por lo tanto disminuiría el valor máximo del producto punto.

El camino óptimo queda limitado a pasar por donde se encuentra el paralelogramo sombreado de la figura III.8 a). Mientras que utilizando el algoritmo de Sakoe-Chiba tenemos disponible todo el rectángulo como se ve en III.8 b).

En general se puede decir que las restricciones utilizadas para la programación dinámica son las siguientes:

1) Puesto que cada palabra comienza con un silencio y termina con otro, el estado inicial en la búsqueda del camino óptimo deberá ser :

MAT[0][0]=S1(0)xS2(0) cuando comienza el silencio en las dos señales.

El estado final en la búsqueda del camino óptimo será:

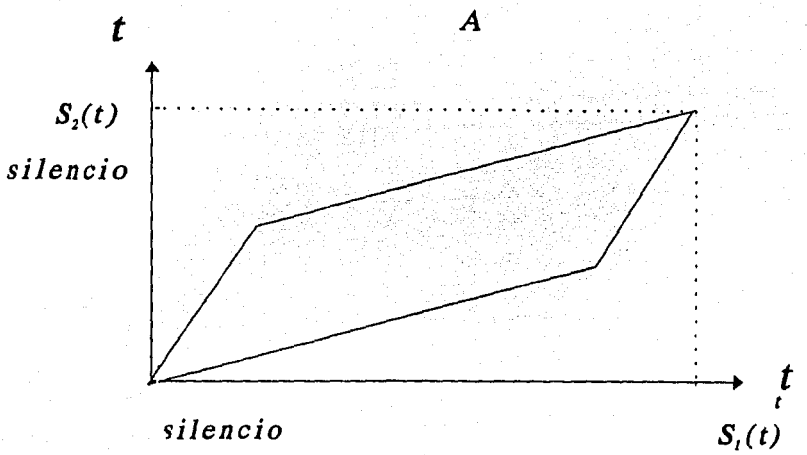
$$MAT[ns1][ns2]=S1(ns1)xS2(ns2)+\text{argmax}\begin{cases}(i,j-1)\\(i-1,j-1)\\(i-1,j)\end{cases}$$

Al recordar que: ns1 es el número de muestras para S_1 y ns2 es el número de muestras para S_2

2) La señal S1(t) se proyectará sobre la señal $S_2(t)$.

ns2 > ns1 Esto implica que se buscará expander la señal $S_1(t)$.

3) El algoritmo para cada casilla de MAT[i][j] será el de Sakoe-Chiba



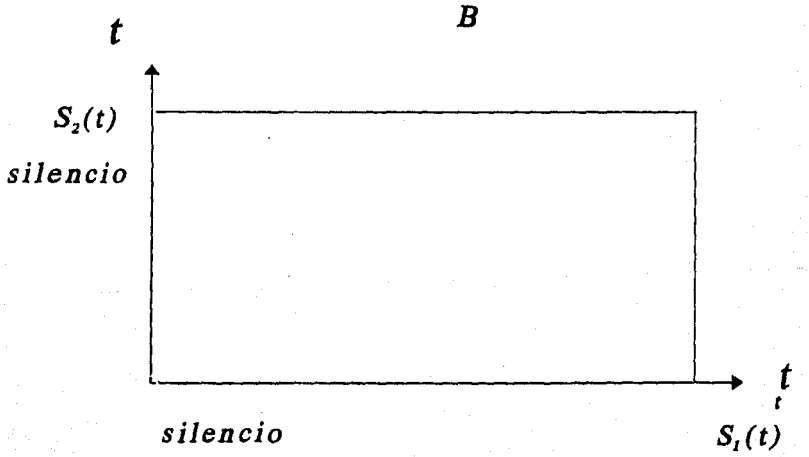


Figura III.8 Diferentes opciones para el camino optimo

4) Continuidad.

No esta permitido, al momento de seleccionar el camino óptimo de hacer saltos mayores de una casilla.

5) Monotonía

Cada muestra de la señal debe ser considerada en el orden natural en que se adquirió.

6) Pendiente

Cada pendiente que se obtenga al final después de haber obtenido el camino óptimo debe ser positiva.

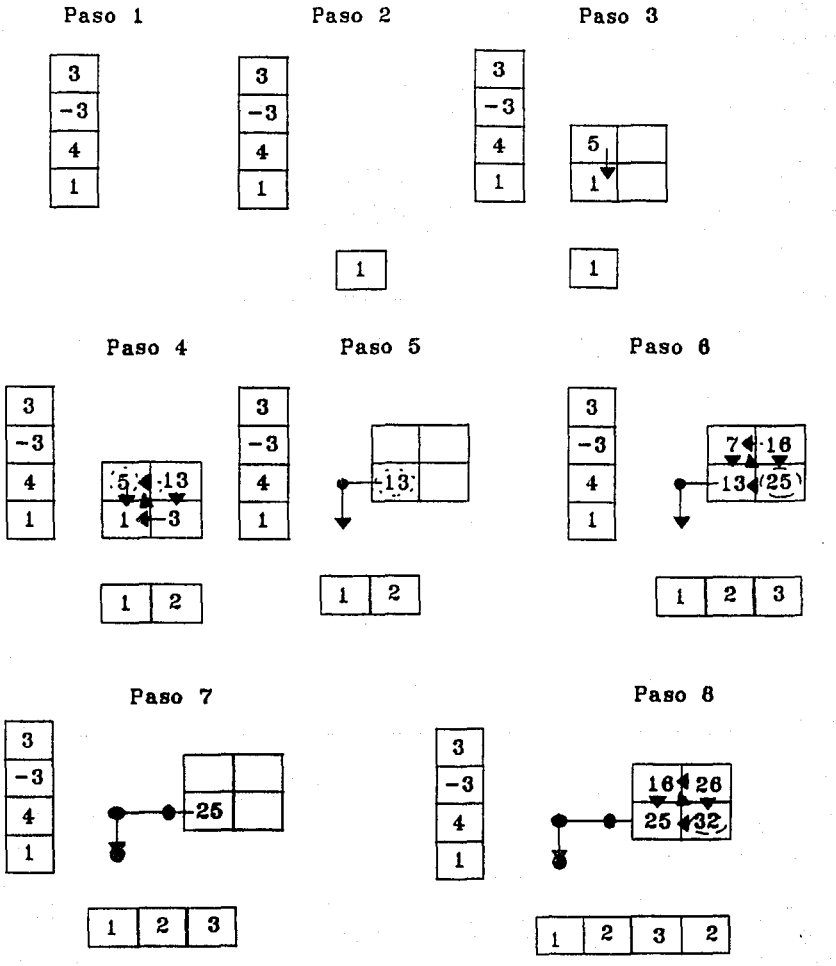
Sin embargo para realizar este método se requiere:

Un número de ns1 x ns2 multiplicaciones
Un número de ns1 x ns2 sumas
Un tamaño de memoria de ns1 x ns2 x 32 bits
Si se piensa que cada archivo de voz posee en promedio 5000 muestras, entonces se necesita
Un número de (5000) x(5000)= 25 000 000 multiplicaciones
Un numero de 24 999 999 sumas
Un tamaño de memoria de 800 000 000 bits = 100 Megabytes

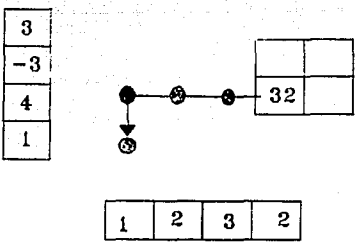
Es decir se necesita un procesador que realice alrededor de 20 MOPS y una memoria de 100 Megabytes. Esto significa que tanto en tiempo como en memoria es inadecuado utilizar el algoritmo de programación dinámica al pie de la letra.

La manera en que se ha implementado la programación dinámica es con una matriz de 2 x 2. Esta matriz, que llamaremos MTD, se ira moviendo conforme avance el camino óptimo. La siguiente figura (III.9) explica en forma gráfica el procedimiento para dos señales de diferente longitud.

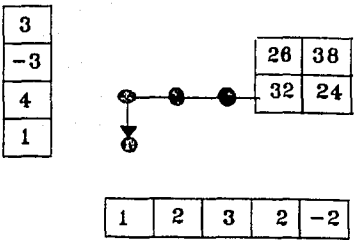
Metodo de Programación Dinámica



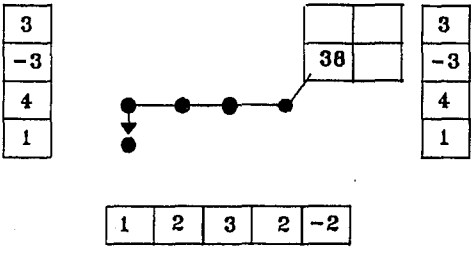
paso 9



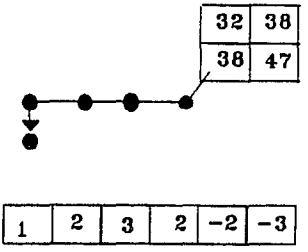
paso 10



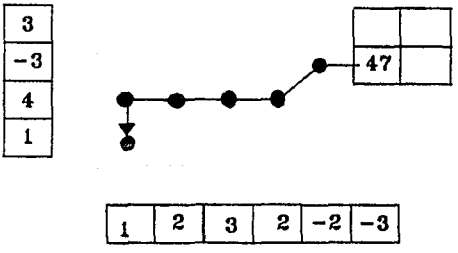
paso 11



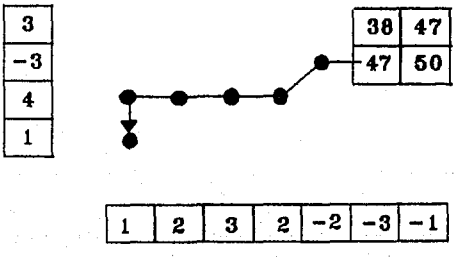
paso 12



paso 13



paso 14



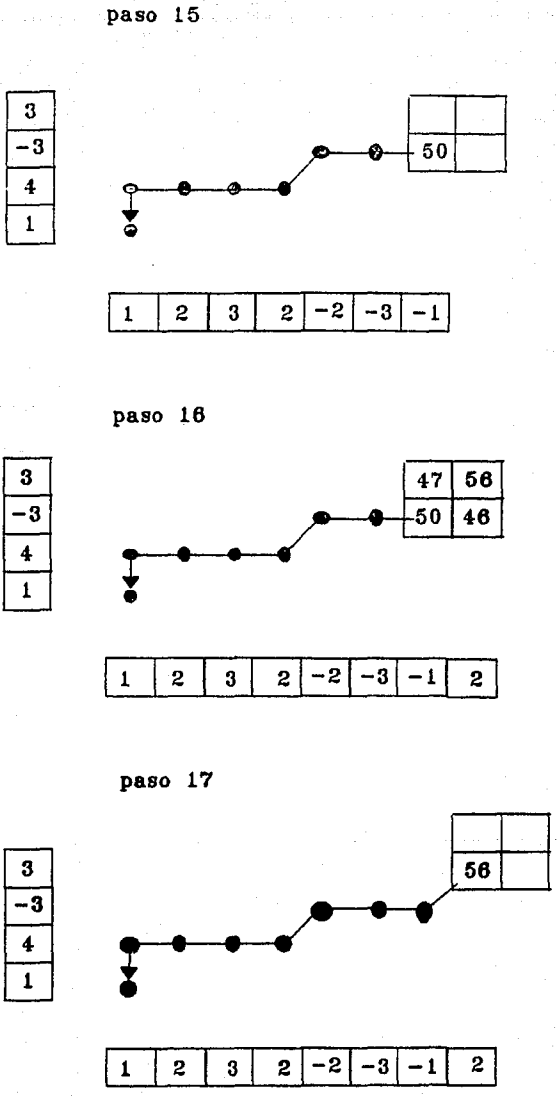


Figura III.9 Método de Programación Dinámica

La segunda función que es necesario definir para llevar a cabo el procedimiento de Gram-Schmidt, es la resta entre dos señales de longitudes diferentes:

Si se recuerda el procedimiento de Gram-Schmidt se aplica la resta a dos señales:

$S_1(t)$ es una señal cualquiera

U1(t) es una señal de energía unitaria

Primero se busca la componente de la proyección de $S_1(t)$ sobre U1(t), sea esta componente c11. Se denota con "o" el producto punto obtenido entre dos señales al utilizar el algoritmo de Sakoe-Chiba de programación dinámica.

c11=S1*(t) o U1*(t)

La señal Ψ será el resultado de la resta:

Ψ1*(t)= S1*(t) - c11U1*(t)

Nuevamente el problema es saber como realizar esta operación con dos señales de diferente longitud.

La solución es utilizar otra vez la programación dinámica. En esta ocasión en vez de buscar el producto punto, se utilizará la energía de la señal resultante.

Se recordará que en el siguiente paso del procedimiento de Gram-Schmidt , se debe encontrar la energía de la señal Y1*(t)

Ey1=integral Y1(t)Y1(t)dt

Si la señal $S_1(t)$ fuera muy parecida a U1(t) entonces el resultado de Y(t) debería de ser muy parecido a cero al igual que su energía.

Esta característica será utilizada como restricción local en la búsqueda del camino óptimo.

En general, se puede decir que sólo tres restricciones cambian respecto del algoritmo anterior.

1) La señal que se encuentra sobre las abscisas tiene que ser unitaria.

2) Cada casilla de la matriz MAT estará determinada por:

MAT[0][0]=/S1(0)-c11U1(0)/

$$MAT[i][j]=/S1(i)-c11U1(j)/ + \text{argmin}\begin{cases}MAT[i][j-1]\\MAT[i-1][j-1]\\MAT[1-1][j]\end{cases}$$

para el primer renglón:

MAT[0][j]=/S1(0)-c11U1(j)/ + MAT[0][j-1]

j=1.....ns2

para la primera columna

MAT[i][0]=/S1(i)-c11Ui(0)/ + MAT[i-1][0]

i=1.....ns1

3) Al ir formando el camino óptimo, la primera casilla que se encuentre al cambiar de columna definirá el elemento que corresponde a Y*(t)

Se toma nuevamente el ejemplo anterior para comprender estas nuevas restricciones:

Sea

$$S_1 = \begin{bmatrix} 1 \\ 4 \\ -3 \\ 2 \end{bmatrix}$$

$$S_2 = \begin{bmatrix} 1 \\ 3 \\ 2 \\ -2 \\ -3 \\ -1 \\ 2 \end{bmatrix}$$

Se busca la energía de la señal S_2

$$E = \sum_{ns2-1}^{i=0} S2^*(t)S2^*(t)$$

E=1+4+9+4+4+9+1+4=36

$e=(36)^{1/2}=6$

U1 es la nueva señal unitaria.

U1=S2(t)/6

$$U1 = \begin{bmatrix} 0.17 \\ 0.33 \\ 0.50 \\ 0.33 \\ -0.33 \\ -0.50 \\ -0.17 \\ 0.33 \end{bmatrix}$$

Nuevamente se utiliza una matriz auxiliar MTD. El procedimiento es el que se observa en la figura III.10. Una vez que se ha determinado una función para realizar el producto punto y otra para realizar la resta en señales de diferente longitud, es posible aplicar completo el método de Gram-Schmidt.

Los diagramas de flujo de las funciones JL_dotp y JL_subs se muestran en el apéndice A, figuras A.4 y A.5 respectivamente.

III.2.6 REALIZACION DEL METODO DE GRAM-SCHMIDT

La parte de entrenamiento dejará listos diez archivos, totalmente limpios de las pausas que tenían antes y después del mensaje, listos para formar la señal ortogonal.

Se supone que se tienen los siguientes archivos de voz con sus respectivas longitudes en número de muestras:

cero	no
uno	n1
dos	n2
tres	n3
cuatro	n4
cinco	n5
seis	n6
siete	n7
ocho	n8
nueve	n9

Como cada señal tiene una longitud diferente, primero hay que indexar estas señales de acuerdo a su longitud.

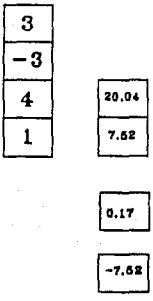
Tomando un ejemplo aleatorio el procedimiento por pasos es como sigue:

Primer paso

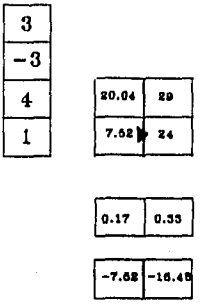
S1 =	dos	ns1 =	n2
S2	cinco	ns2	n5
S3	tres	ns3	n3
S4	ocho	ns4	n8
S5	uno	ns5	n1
S6	nueve	ns6	n9
S7	siete	ns7	n7
S8	seis	ns6	n6
S9	cuatro	ns9	n4
S10	cero	ns10	n0

Programación Dinámica en la resta

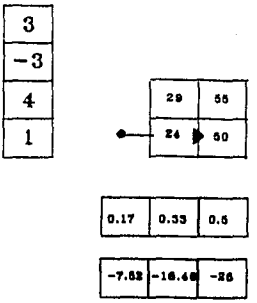
Paso 1



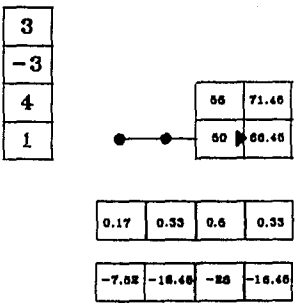
Paso 2



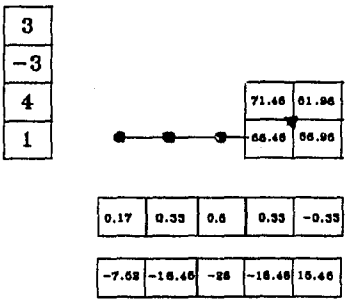
Paso 3



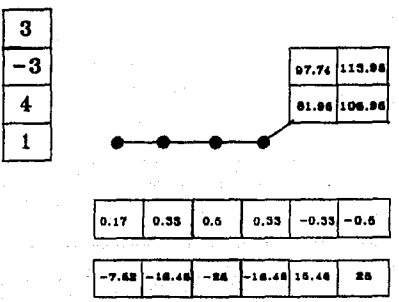
Paso 4



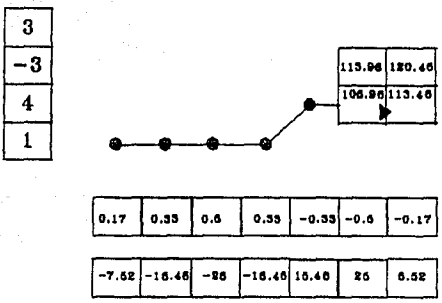
Paso 5



Paso 6



Paso 7



Paso 8

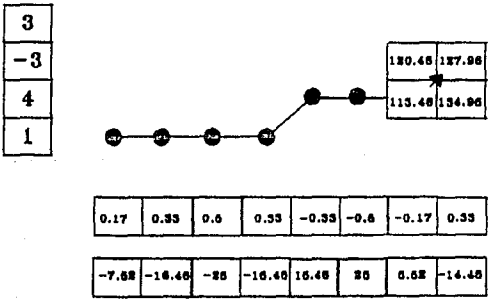


Figura III.10
Procedimiento para realizar la resta entre dos señales

Segundo paso

Una matriz MI guardará el resultado de este orden

$$MI=\begin{bmatrix}2\\5\\3\\8\\1\\9\\7\\6\\4\\0\end{bmatrix}\qquad\begin{bmatrix}n2\\n5\\n3\\n8\\n1\\n9\\n7\\n6\\n4\\n0\end{bmatrix}$$

Tercer paso

Crear una matriz MC que guarde el resultado de cada proyección:

MC es de dimensión 10 x 10

Cuarto paso

Calcular MC[0][0]

Sea

$$E=\sum_{i=0}^{ns1-1} S1^*(t) \times S1^*(t)$$

$$MC[0][0]= (E)^{1/2}$$

Quinto paso

Calcular U1*(t) , primera señal de energía unitaria

$$U1^*(t)=S1^*(t)/(E)^{1/2}$$

Sexto paso

Proyectar S2*(t) sobre U1*(t)

MC[1][0]=S2*(t) o U1*(t)

Séptimo paso

Encontrar la componente perpendicular , esta se llamará Ψ

$\Psi = S2^*(t) - MC[1][0]\ U1^*(t)$

Ψ tendrá la misma longitud que U1

Octavo paso

El resultado de Ψ hacerlo ortogonal

$$E = \sum_{i=0}^{ns1-1} \Psi1^*(t) \times \Psi1^*(t)$$

$U2^*(t) = \Psi/(E)^{1/2}$

$MC[1][1] = (E)^{1/2}$

A partir de este momento, se repite la secuencia desde el paso número 7.

Noveno paso

Proyectar S3(t) sobre U1(t) y U2(t)

MC[2][0]=S3*(t) o U1*(t)

MC[2][1]=S3*(t) o U2*(t)

Décimo paso

Encontrar la componente perpendicular a las dos señales anteriores.

$\Psi = S3^*(t) - MC[2][0]U1^*(t) + MC[2][1]U2^*(t)$

la longitud de Ψ es de ns1

Undécimo paso

Hacer ortogonal la señal resultante Ψ

$$E = \sum_{i=0}^{ns1-1} \Psi1^*(t) \times \Psi1^*(t)$$

$U3^*(t)=\Psi/(E)^{1/2}$

$MC[2][2] = (E)^{1/2}$

Duodécimo paso

Proyectar S4*(t) sobre U1*(t), U2*(t), U3*(t)

MC[3][0]= S4 o U1

MC[3][1]= S4 o U2

MC[3][2]= S4 o U3

Décimotercer paso

Encontrar la componente perpendicular a las señales anteriores

Ψ=S4-MC[3][0]U1 - MC[3][1]U2 - MC[3][2]U3

Décimocuarto paso

hacer la señal resultante ortogonal

$$E=\sum_{i=0}^{ns1-1} \Psi 1^*(t) \times \Psi 1^*(t)$$

$U4^*(t)=\Psi 1/(E)^{1/2}$

$MC[3][3] = (E)^{1/2}$

Este proceso se repite hasta terminar con la señal S9.

El diagrama de flujo para este procedimiento se muestra en la figura A.6 del apéndice A.

III.2.7 RECONOCIMIENTO CON EL METODO DE GRAM-SCHMIDT

Una vez que se ha construido el espacio vectorial en la fase de reconocimiento, se vuelve a grabar una señal, aquélla que se va a reconocer.

La señal debe sufrir el mismo procedimiento que cualquier señal en la fase de entrenamiento. La señal pasa por la función JL_qb en donde se suprimen las pausas que existen antes y después de la palabra.

Nuevamente se utilizará programación dinámica para obtener las componentes de esta señal en cada uno de los ejes del espacio vectorial.

Un vector MR[10] guardará cada uno de las componentes de la señal. El procedimiento es como sigue:

MR[0]= S(t) o U0(t)

MR[1]=S(t) o U1(t)

MR[2]=S(t) o U2(t)

.
.
.

MR[9]=S(t) o U9(t)

Una vez que el vector MR ha sido formado, se busca la mínima distorsión con el vector renglón de la matriz MC que contiene todos los coeficientes de proyección que se realizan en la fase de entrenamiento.

El error de distorsión se define ahora en términos de los coeficientes de proyección:

$$e(i)=\sum_{j=0}^{9} MC[i][j] - MR[j]$$

El vector e(i) contendrá la distorsión que existe entre la señal que entró y cada una de las señales patrón.

El elemento mínimo de e(i) dirá qué señal se parece más a las que han sido registradas anteriormente. Sin embargo, del elemento mínimo de e(i) sólo interesa saber su índice. Este dato se utilizará con la matriz MI entonces MI[i][0] es el dígito seleccionado.

Este procedimiento se encuentra de manera gráfica en la figura III.11 .

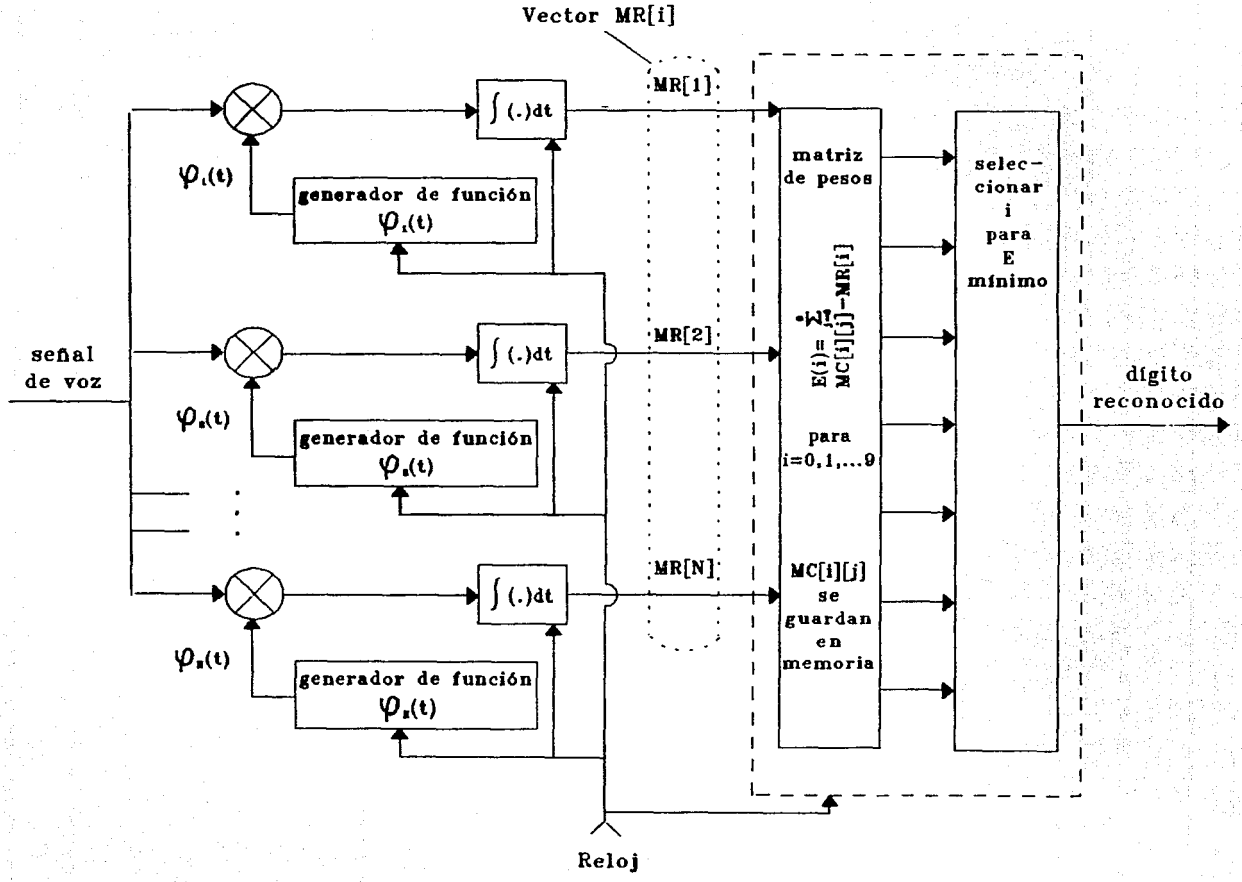


Figura III.11 Metodo de reconocimiento

RESULTADOS Y CONCLUSIONES

El programa de reconocimiento se llevó acabo para una serie de diez números siendo éstos del cero al nueve. El sistema es para un reconocedor de palabras aisladas monolocutor.

El programa se realizó con el procedimiento de Gram-Schmidt y utiliza una alineación en tiempo dada por programación dinámica.

Sin embargo, la eficiencia del reconocedor es baja. De cada cinco intentos sólo uno es reconocido.

Para considerar cuáles son los parámetros que influyen en este rendimiento se podría seguir el proceso que sufre toda la señal, ya sea que ésta se utilice para formar el conjunto patrón de señales ortogonales, o bien, sea la palabra a reconocer.

La primera parte del proceso ocurre al hacer la toma de muestra. Este primer paso puede ser decisivo para construir todo el reconocedor.

Dos parámetros son importantes para tomarse en cuenta:

1) La velocidad de muestreo.

2) El tiempo durante el cual se toma la muestra.

Para el primero es importante considerar que existen algunos fonemas que pueden ir más allá de los 10 kHz; muchas veces cuando se graba voz de alta fidelidad, el muestreo se hace a 15 kHz.

De esta manera es necesario tener un mínimo de 8 kHz de muestreo (Esta es la medida estándar). El reconocedor de palabras aisladas que se propone se probó con una frecuencia de muestreo de 7 kHz en una ocasión, el resultado fue que de diez intentos de reconocimiento ninguno fue exitoso.

Se debe tener en mente que aunque existen partes muy redundantes en la voz existen otras que pueden proporcionar información vital. Debido a esto es necesario buscar la mejor calidad de la señal con una velocidad de muestreo alta o cuando menos de 8 kHz.

Durante el primer capítulo de la introducción se habló de las cuestiones de prosodia que se pueden presentar en la voz. Es posible que existan personas que hablen muy rápido y en un segundo o dos puedan decir algunas palabras completas. Sin embargo, también se debe de considerar que pueden existir personas que pueden hablar sumamente lento. Si éste es el caso, necesitarán de más tiempo para expresar una palabra. Si en el reconocedor existe un tiempo limitado para que hable la persona, entonces cabe la posibilidad de que la palabra quede cortada y por lo tanto la información para reconocer estará muy distorsionada.

Este problema se presenta en general con dígitos cuya pronunciación es larga como la del siete, el seis o el cinco.

En ambos casos lo ideal sería tener una velocidad de muestreo alta y tomar un número suficiente de muestras para hacer el análisis de reconocimiento. El problema que se presenta es la limitación de memoria del reconocedor.

Como se mencionó, en algún momento se requiere una cantidad de memoria considerable para poder hacer cálculos confiables. Cada señal es de aproximadamente cinco mil muestras y cada muestra es de treinta y dos bits.

En el caso de que la memoria del reconocedor sea muy limitada, es posible intentar una compresión de la señal de voz. Sin embargo, antes de realizarla es necesario eliminar las pausas del principio y del final de la palabra que no contienen realmente información.

Si la señal estuviera limpia de pausas, sería muy conveniente aplicar una compresión que funcione por correlación:

a) Las partes de señal que tengan una frecuencia alta poseen información valiosa y por tanto no serán comprimidas.

b) Las partes de señal con una frecuencia muy lenta poseen información redundante y pueden ser comprimidas.

En la realización de este programa el filtrado de la señal se hace sobre las pausas que existen al principio y al final.

Sin embargo, por restricciones de memoria nos vemos obligados a quitar muestras en algunas pausas intermedias de alguna palabra. Por ejemplo, se-is o si-e-te u o-ch-o tienen cierta discontinuidad que los caracteriza y que es importante para su reconocimiento.

Como la toma de decisión para hacer el filtrado está basada en cuestiones de energía de la señal; es decir, en amplitudes de la señal, entonces aquellas muestras que son más pequeñas en amplitud son las más suceptibles de que sean borradas por este algoritmo de decisión.

Cuando ocurre ésto en señales de voz muy largas, como las que mencionamos anteriormente, entonces se pierde mucha información vital. Un "dos" puede entonces parecerse a un "ocho" o a un "cinco".

ocho
dos
cinco

Al existir menos pausas es más fácil que el reconocedor se confunda. En algunos casos, la medida de la distorsión para algunas señales y los archivos patrones, que ya han sido grabados previamente, es prácticamente igual.

Nuevamente el "dos" se parece al "ocho" o al "cero". En una prueba se obtuvo como medida de distorsión: 93.456, 96.664 y 97.438, es decir, que prácticamente no existía diferencia entre las señales.

De aquí la gran importancia de seleccionar correctamente las señales que serán utilizadas para formar el espacio vectorial.

Se toma como ejemplo el número "ocho" antes de quitársele las pausas y cuando ya no las tiene. En resultados experimentales sin la pausa intermedia que separa las dos "o" en "o-ch-o", es muy fácil confundirse con un dos por ejemplo.

Cuando se genera el espacio vectorial, el segundo factor importante es la programación dinámica. Se podría pensar que el hecho de maximizar el producto punto puede afectar para cuando existe una gran discrepancia entre los tamaños de algunas palabras.

Tomemos el caso del número "dos". Esta palabra es muy rápida de pronunciar; sin embargo, el sistema llega a confundirse y coloca una proyección mayor sobre las señales que tienen una longitud pequeña.

Para observar ésto se presentan tres señales: Un "dos", un "cinco", que fueron utilizados para generar el espacio vectorial, y un "dos" que fue la señal a reconocer. El sistema dijo que era un "cinco".

Aunque el rendimiento del reconocedor sea bajo, es necesario darse cuenta que aún pueden implementarse algunas mejoras. Un crecimiento de memoria sería lo más adecuado.

De cualquier manera, se puede decir que cada vez estamos más cerca de llegar a reconocer y comprender la palabra humana por medios artificiales.

APENDICE A

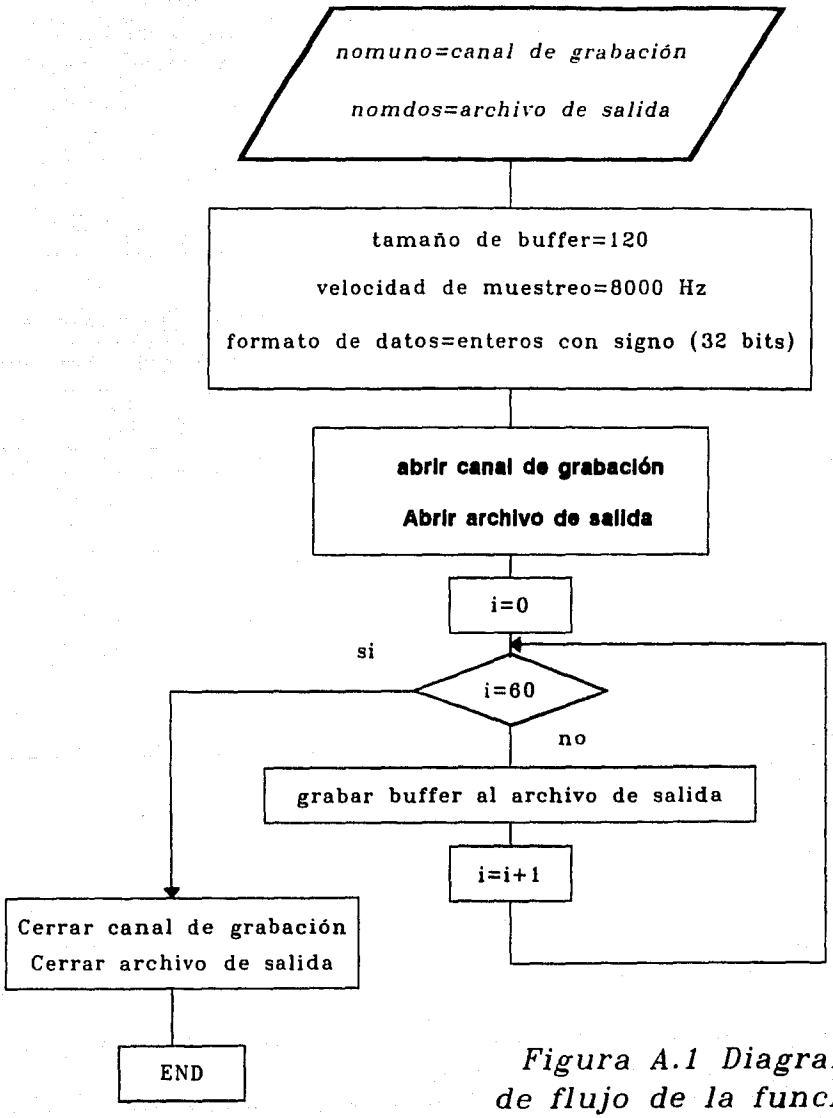
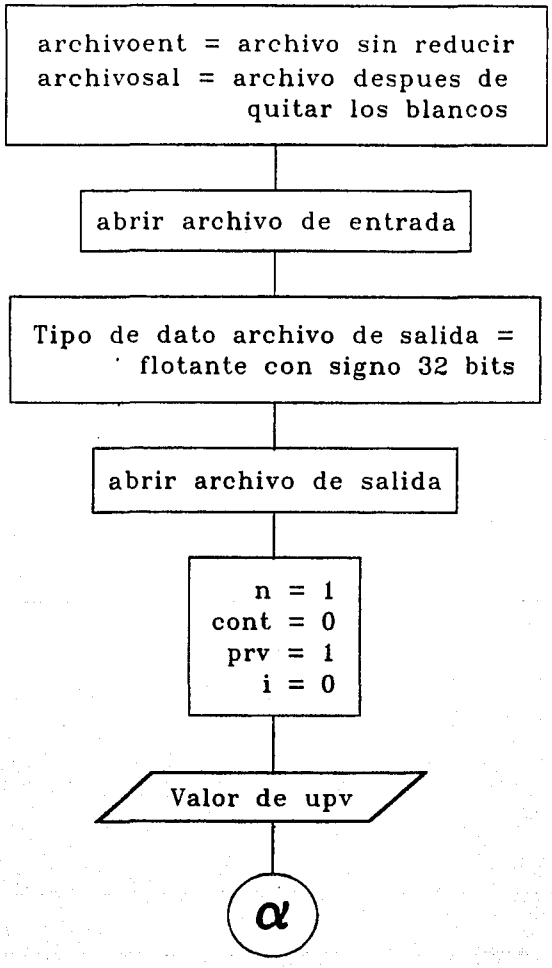
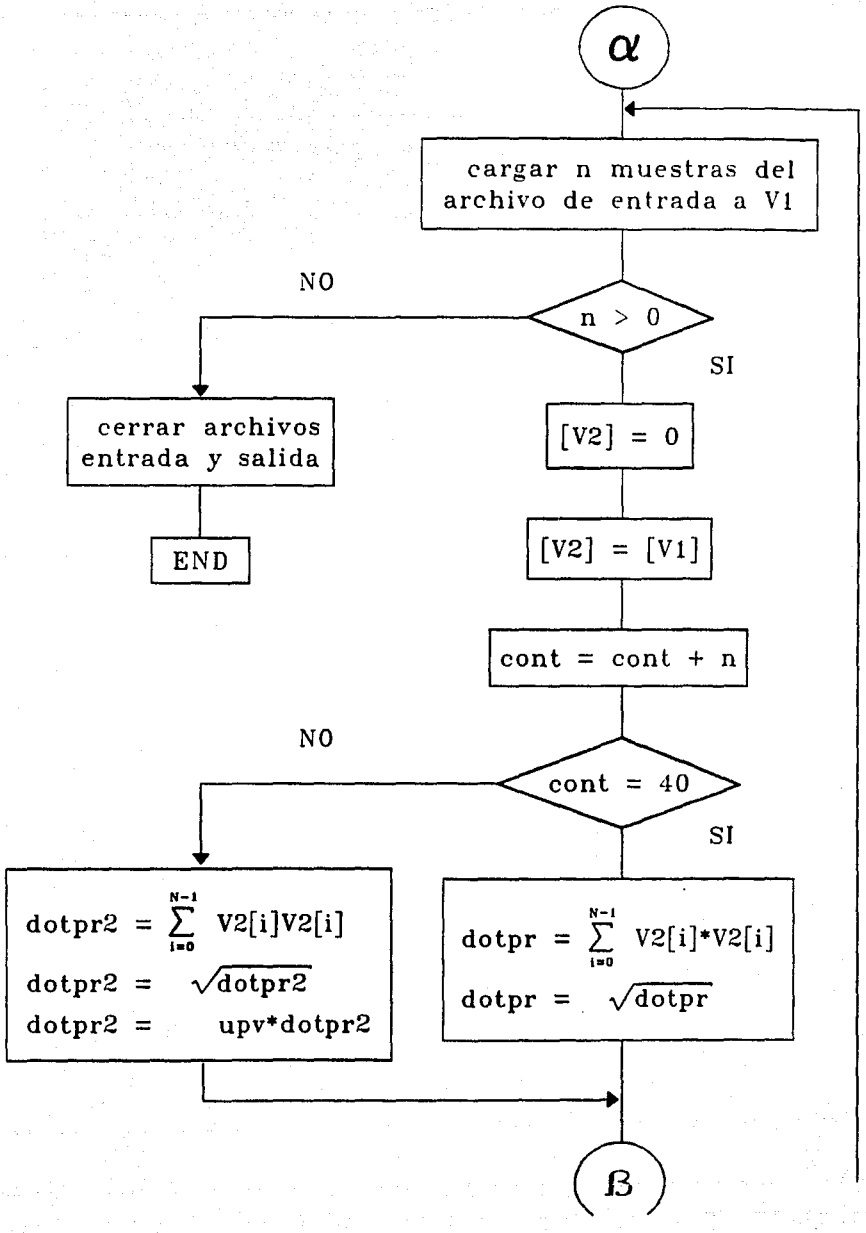


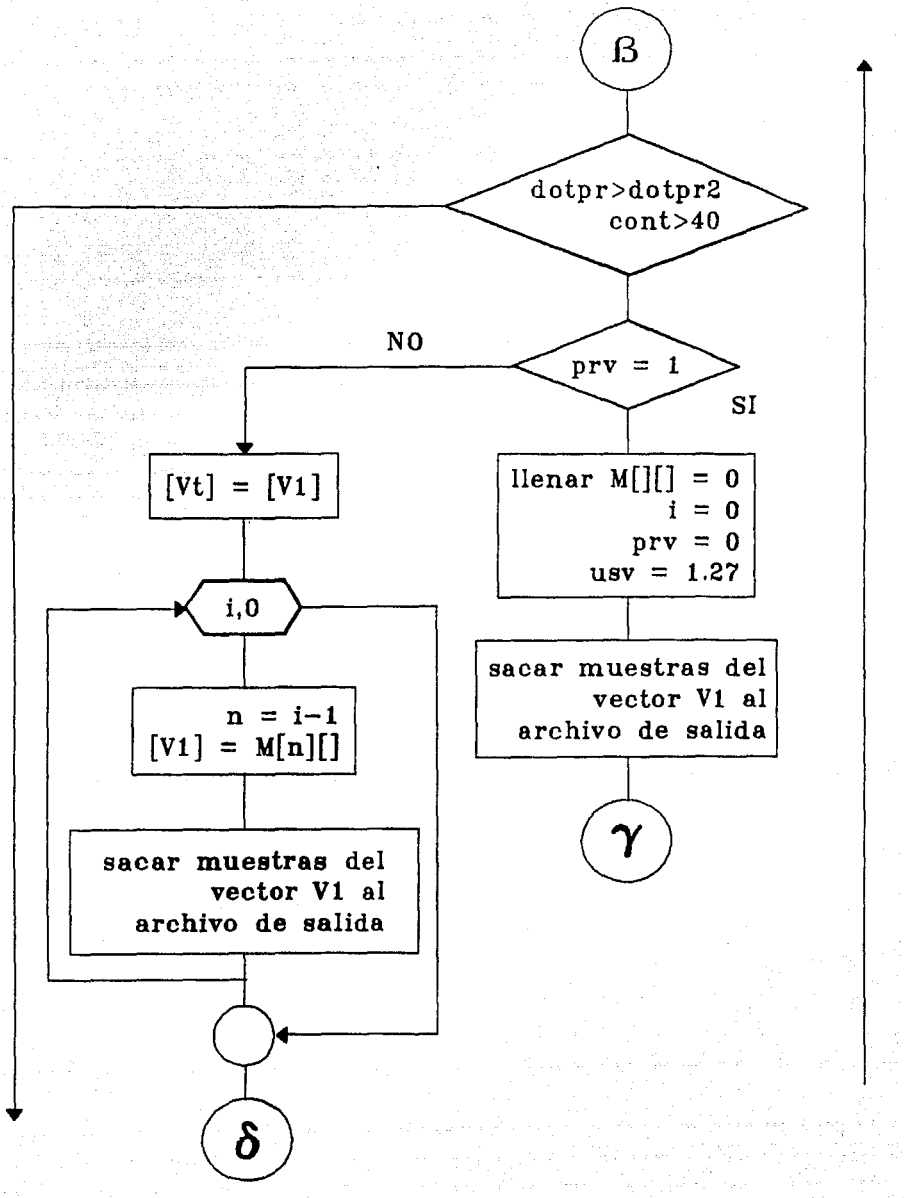
Figura A.1 Diagrama de flujo de la función JL faq

QD3H.H

Void JL_qb (char archivoent [15], char archivosal [15])




α
cargar n muestras del archivo de entrada a V1
NO
n > 0
SI
cerrar archivos entrada y salida
END
[V2] = 0
[V2] = [V1]
cont = cont + n
NO
cont = 40
SI
dotpr2 = Σ V2[i]V2[i]
dotpr2 = √dotpr2
dotpr2 = upv*dotpr2
dotpr = Σ V2[i]*V2[i]
dotpr = √dotpr
β


ß
dotpr>dotpr2
cont>40
NO
prv = 1
SI
[Vt] = [V1]
llenar M[][] = 0
i = 0
prv = 0
usv = 1.27
i,0
sacar muestras del
vector V1 al
archivo de salida
n = i-1
[V1] = M[n][]
sacar muestras del
vector V1 al
archivo de salida
γ
δ

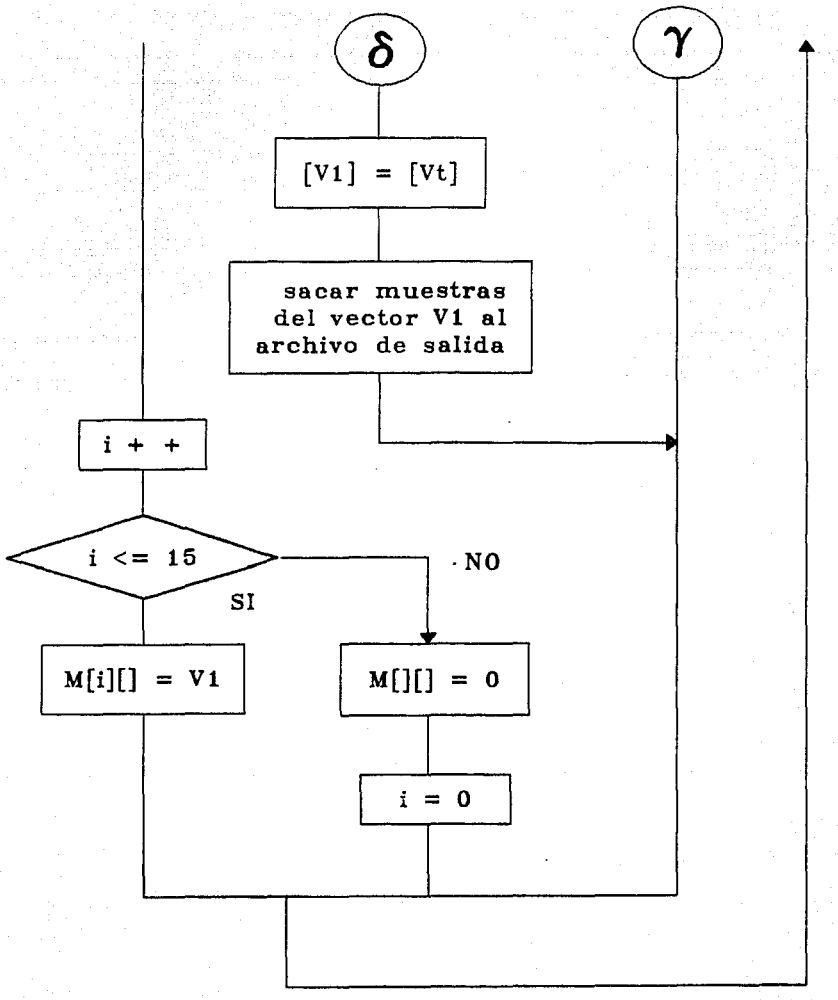
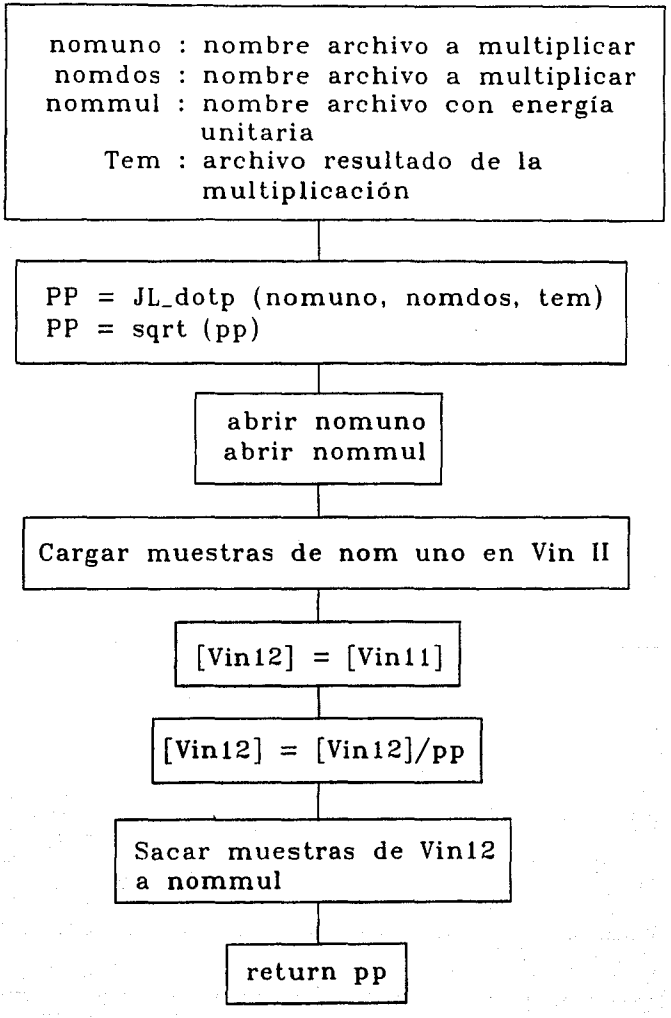


Figura A.2 Diagrama de flujo de la función JL_qb

Acosh.h

float JL_cos (char nomuno[8], char nomdos[8]
char nommul [8], char tem[3])



Acosph.h

float JL_cosp (char nomuno [8], charnomdos [8], char nomres [8], char tem [3])

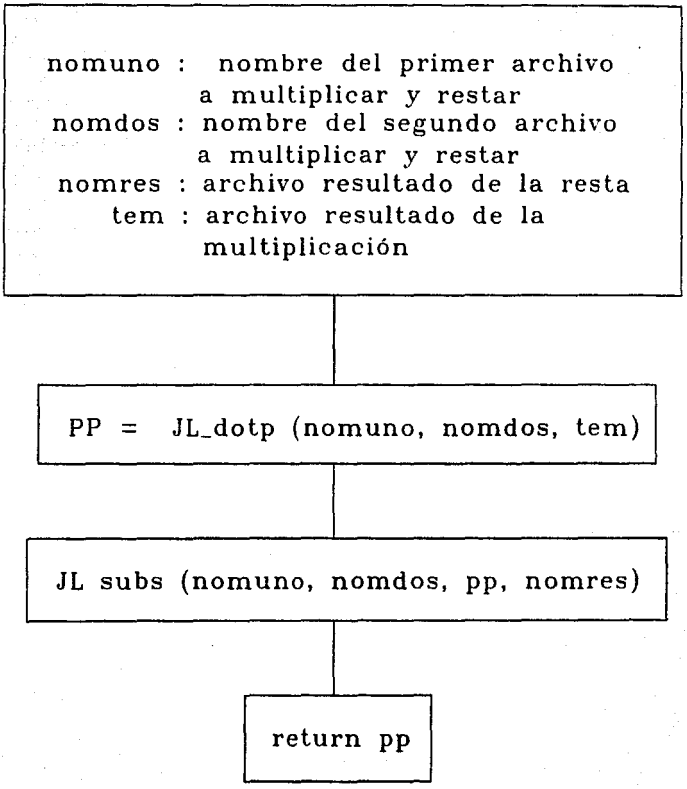
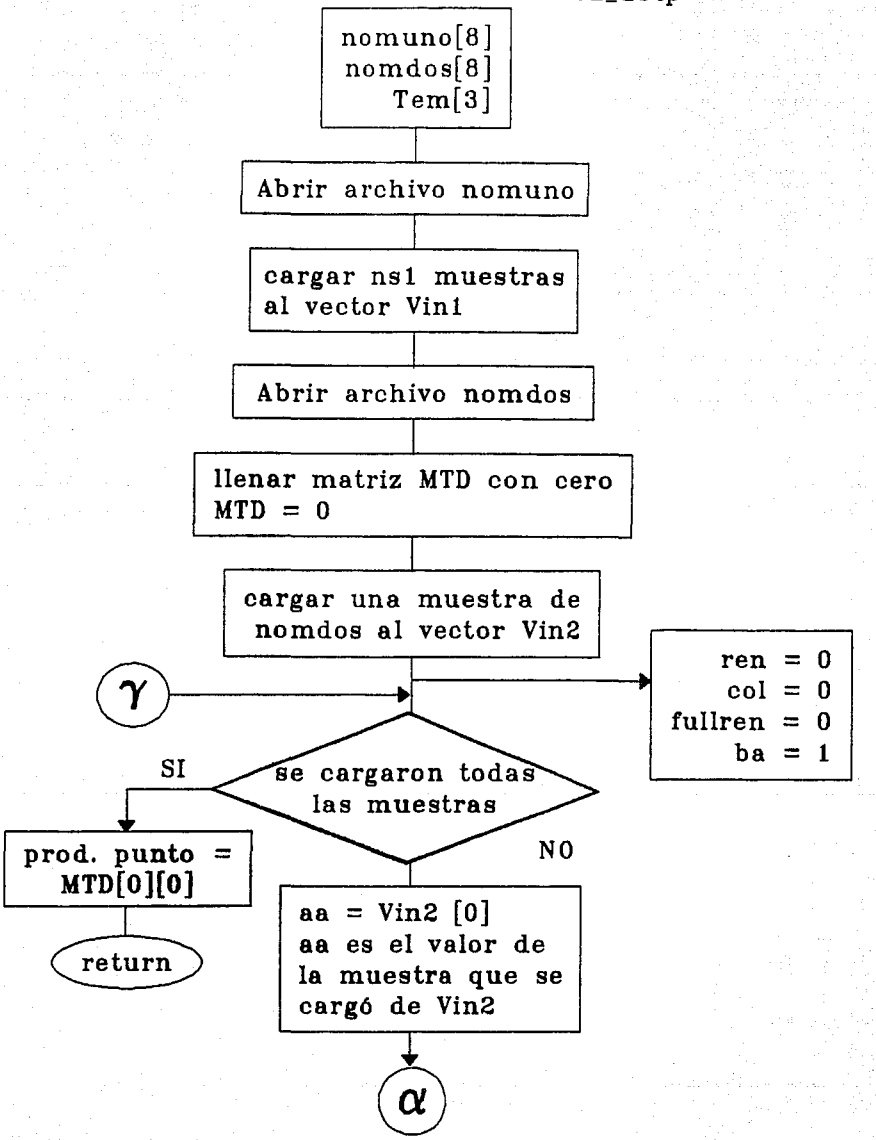
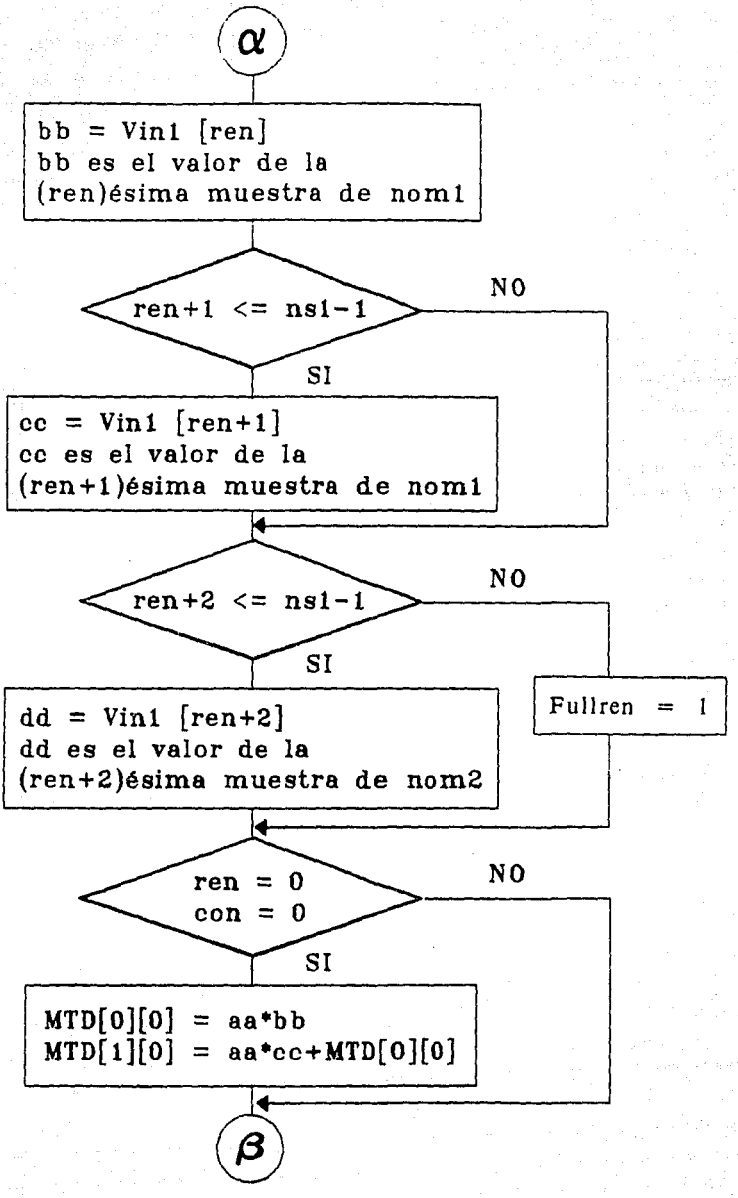


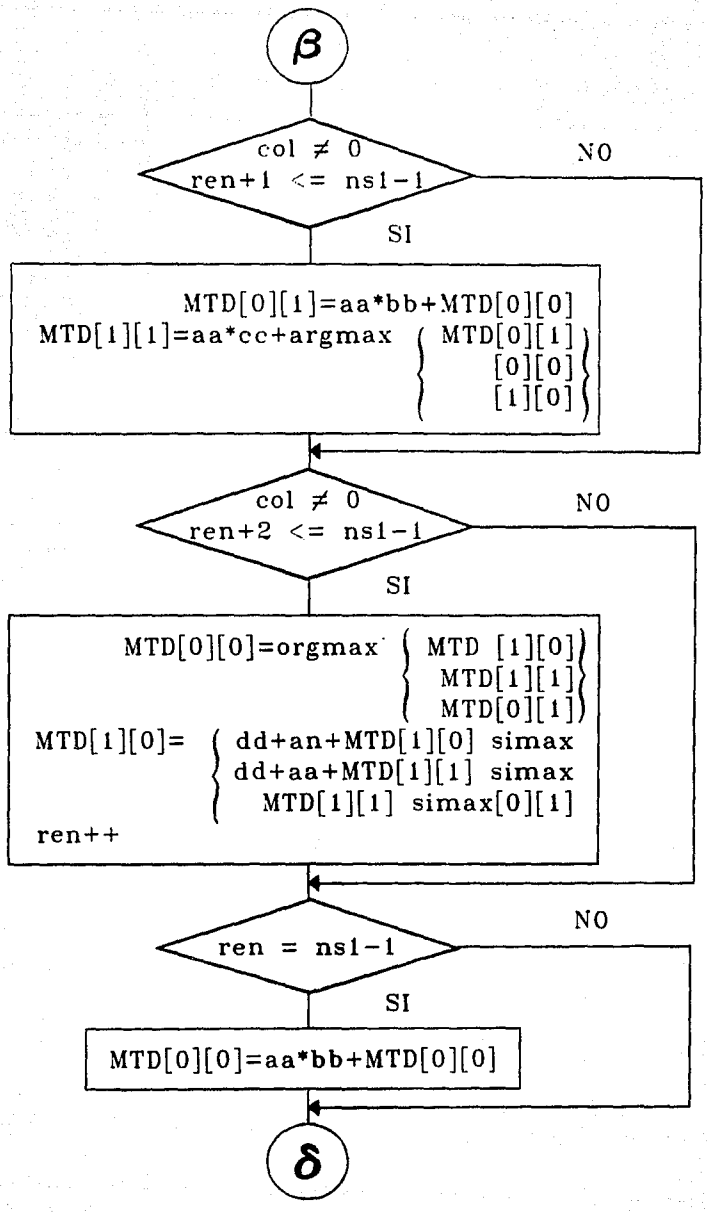
Figura A.3 Programa que genera el espacio vectorial

APP2H.H

JL_dotp




α
bb = Vin1 [ren]
bb es el valor de la
(ren)ésima muestra de nom1
ren+1 <= ns1-1
NO
SI
cc = Vin1 [ren+1]
cc es el valor de la
(ren+1)ésima muestra de nom1
ren+2 <= ns1-1
NO
SI
dd = Vin1 [ren+2]
dd es el valor de la
(ren+2)ésima muestra de nom2
Fullren = 1
ren = 0
con = 0
NO
SI
MTD[0][0] = aa*bb
MTD[1][0] = aa*cc+MTD[0][0]
β


β
col ≠ 0
ren+1 <= ns1-1
NO
SI
MTD[0][1]=aa*bb+MTD[0][0]
MTD[1][1]=aa*cc+argmax { MTD[0][1], [0][0], [1][0] }
col ≠ 0
ren+2 <= ns1-1
NO
SI
MTD[0][0]=orgmax { MTD [1][0], MTD[1][1], MTD[0][1] }
MTD[1][0]= { dd+an+MTD[1][0] simax, dd+aa+MTD[1][1] simax, MTD[1][1] simax[0][1] }
ren++
ren = ns1-1
NO
SI
MTD[0][0]=aa*bb+MTD[0][0]
δ

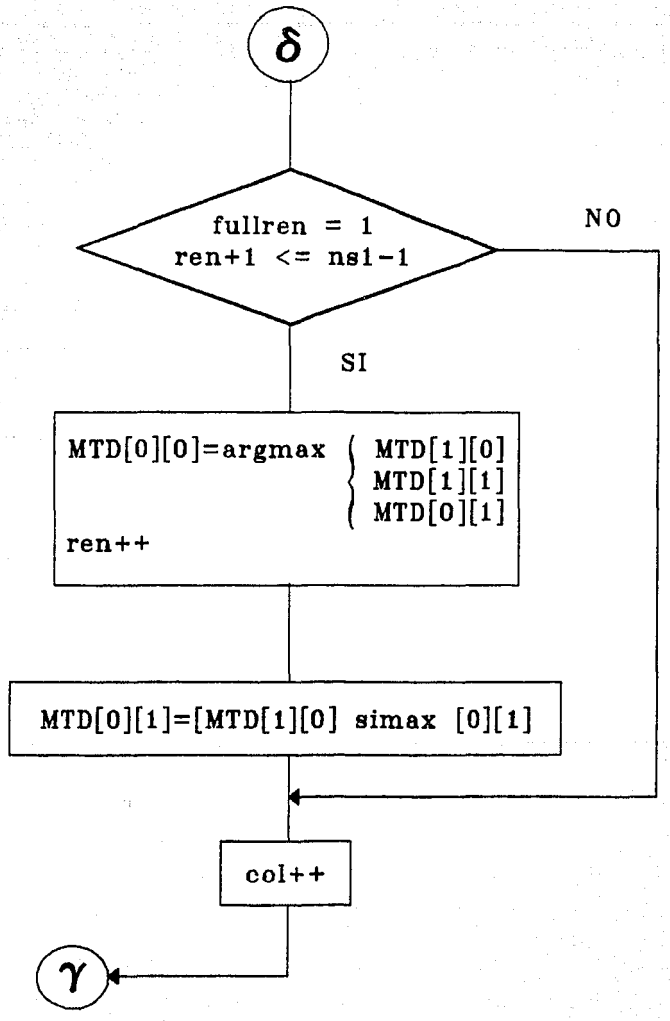
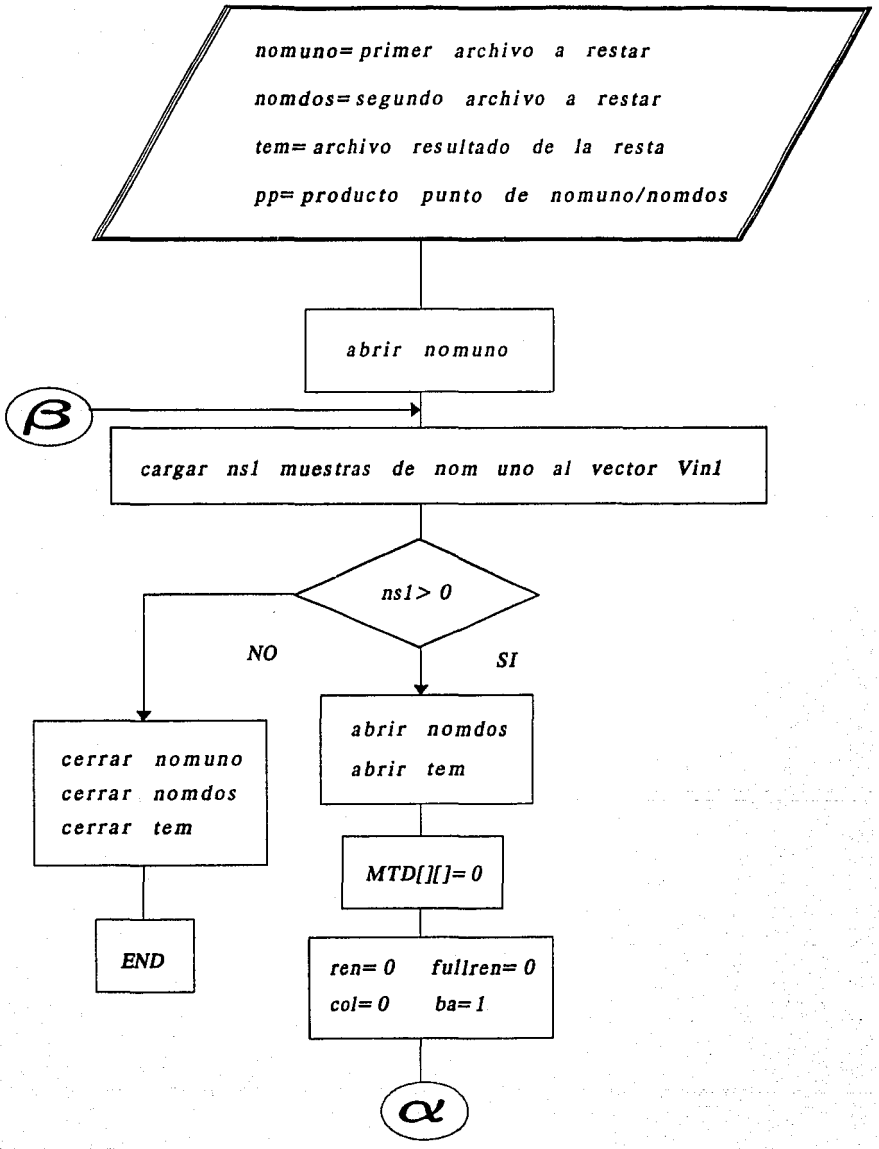
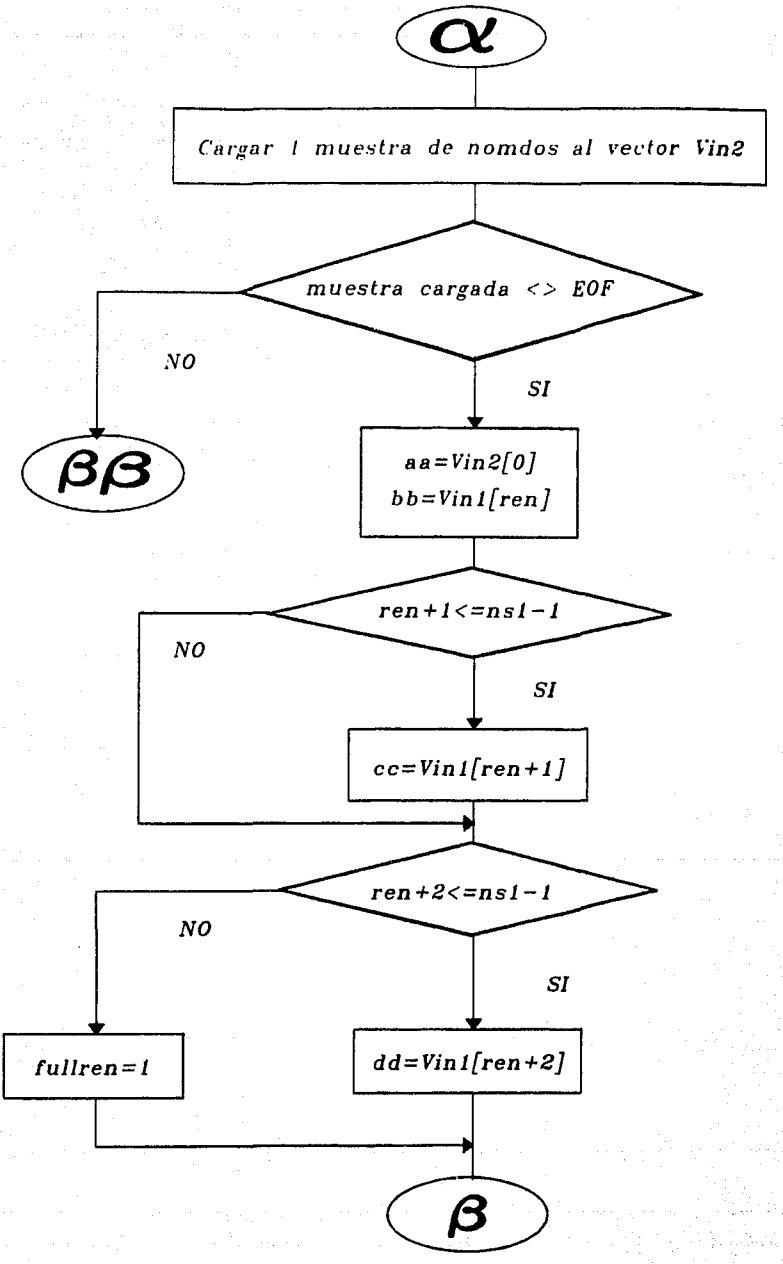
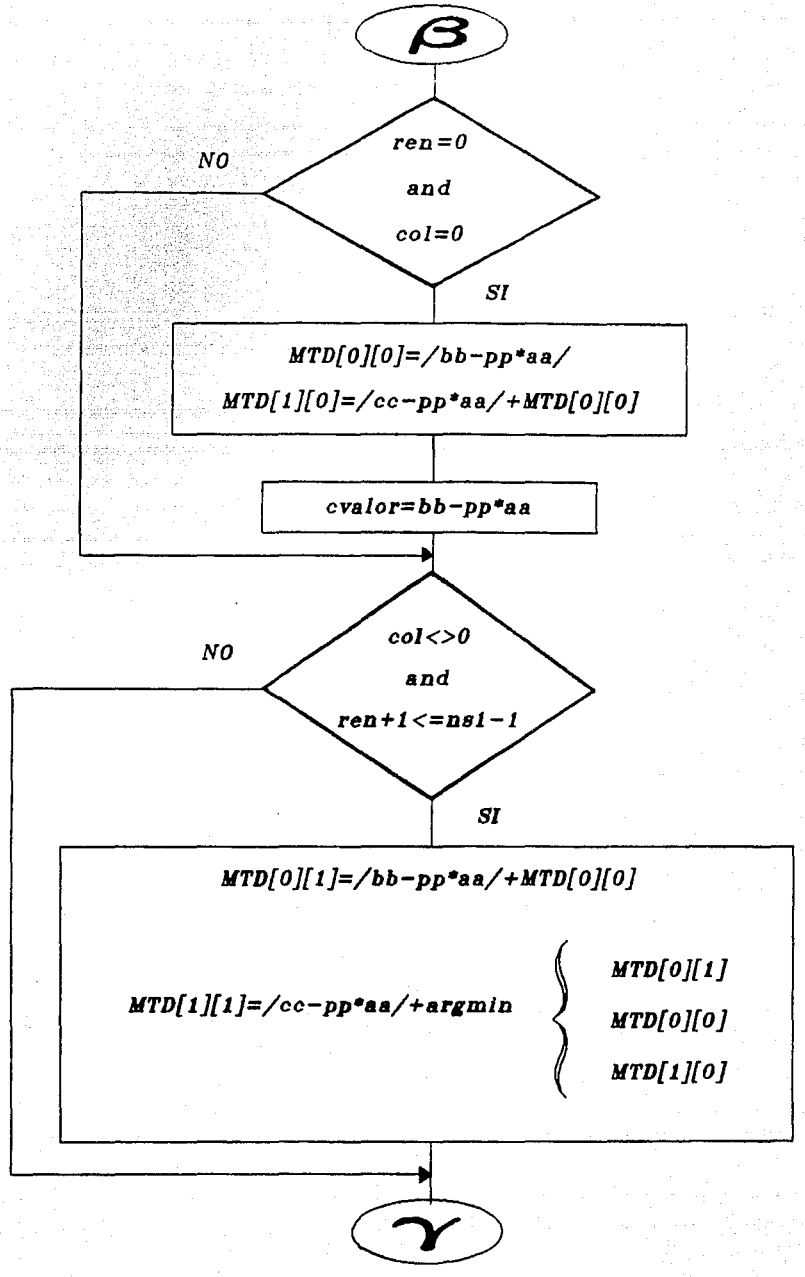
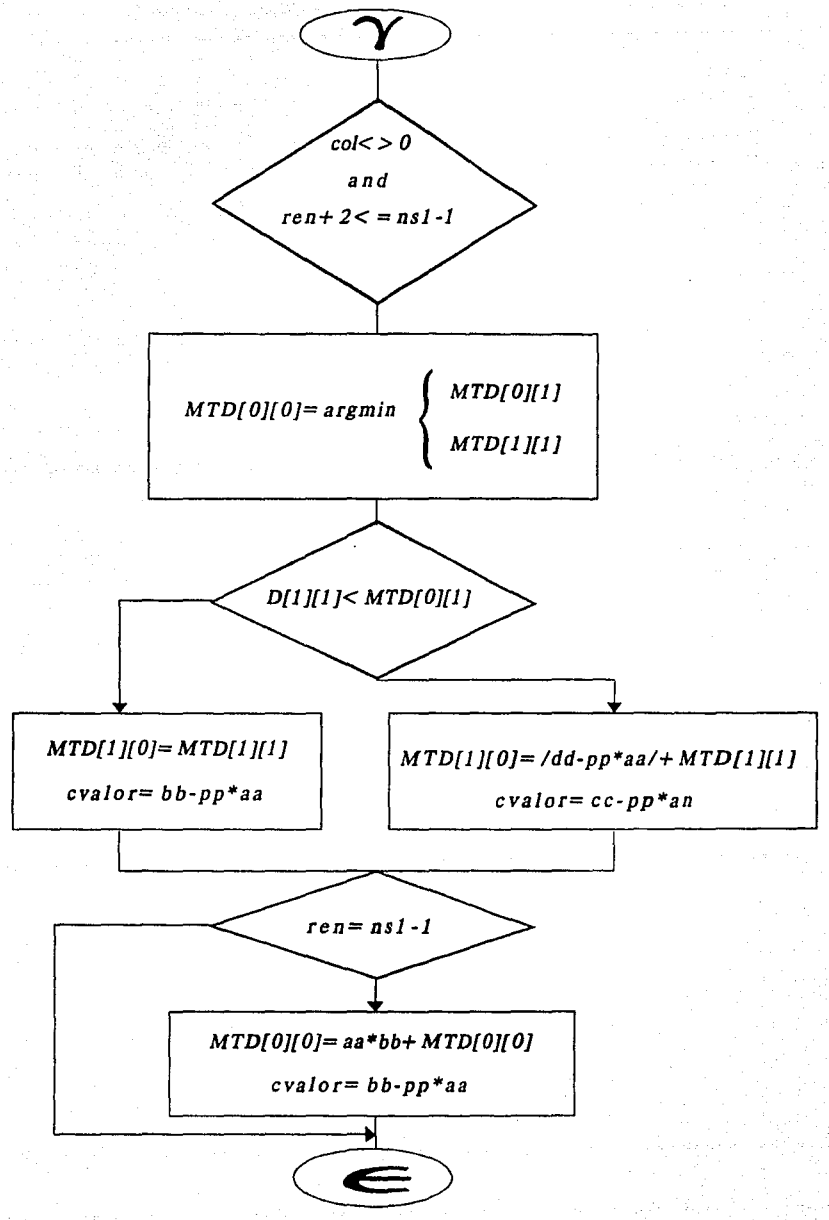


Figura A.4 Diagrama de flujo de APP2H.H


nomuno= primer archivo a restar
nomdos= segundo archivo a restar
tem= archivo resultado de la resta
pp= producto punto de nomuno/nomdos
abrir nomuno
β
cargar ns1 muestras de nom uno al vector Vin1
ns1> 0
NO
SI
cerrar nomuno
cerrar nomdos
cerrar tem
END
abrir nomdos
abrir tem
MTD[][]= 0
ren= 0 fullren= 0
col= 0 ba= 1
α


α
Cargar 1 muestra de nomdos al vector Vin2
muestra cargada <> EOF
NO
SI
ββ
aa=Vin2[0]
bb=Vin1[ren]
ren+1<=ns1-1
NO
SI
cc=Vin1[ren+1]
ren+2<=ns1-1
NO
SI
fullren=1
dd=Vin1[ren+2]
β


B
ren=0
and
col=0
NO
SI
MTD[0][0]=/bb-pp*aa/
MTD[1][0]=/cc-pp*aa/+MTD[0][0]
cvalor=bb-pp*aa
col<>0
and
ren+1<=ns1-1
NO
SI
MTD[0][1]=/bb-pp*aa/+MTD[0][0]
MTD[1][1]=/cc-pp*aa/+argmin
MTD[0][1]
MTD[0][0]
MTD[1][0]
γ


col< > 0
and
ren+ 2< = ns1 -1
MTD[0][0]= argmin
MTD[0][1]
MTD[1][1]
D[1][1]< MTD[0][1]
MTD[1][0]= MTD[1][1]
cvalor= bb-pp*aa
MTD[1][0]= /dd-pp*aa/+ MTD[1][1]
cvalor= cc-pp*an
ren= ns1 -1
MTD[0][0]= aa*bb+ MTD[0][0]
cvalor= bb-pp*aa

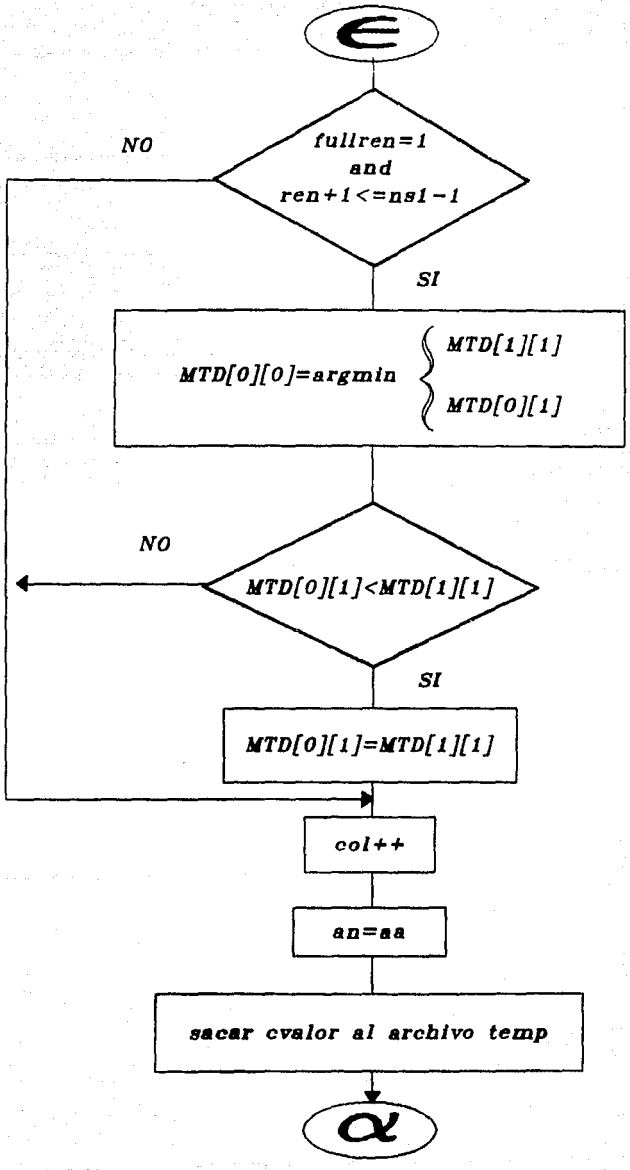
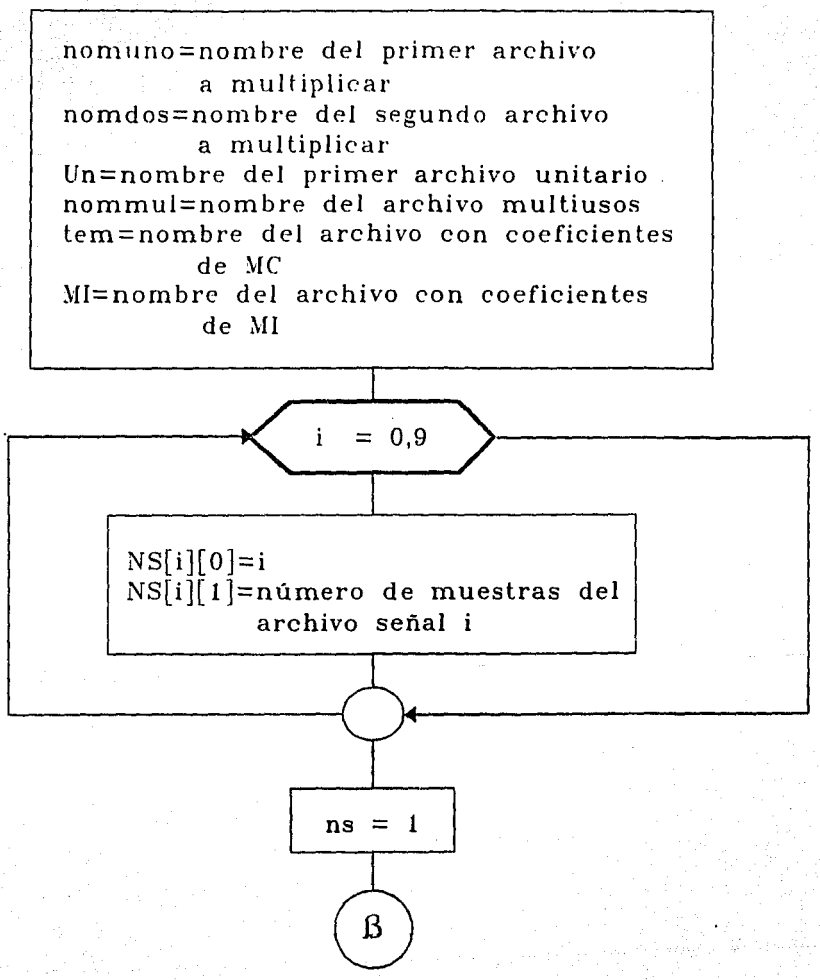
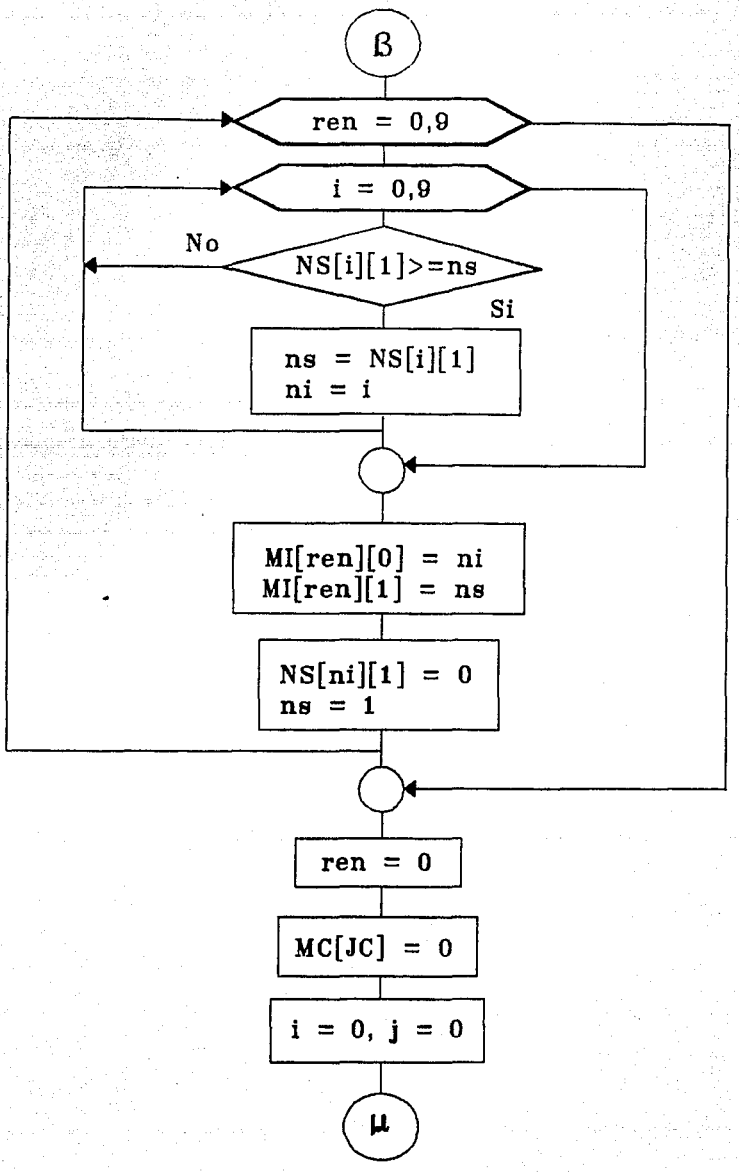
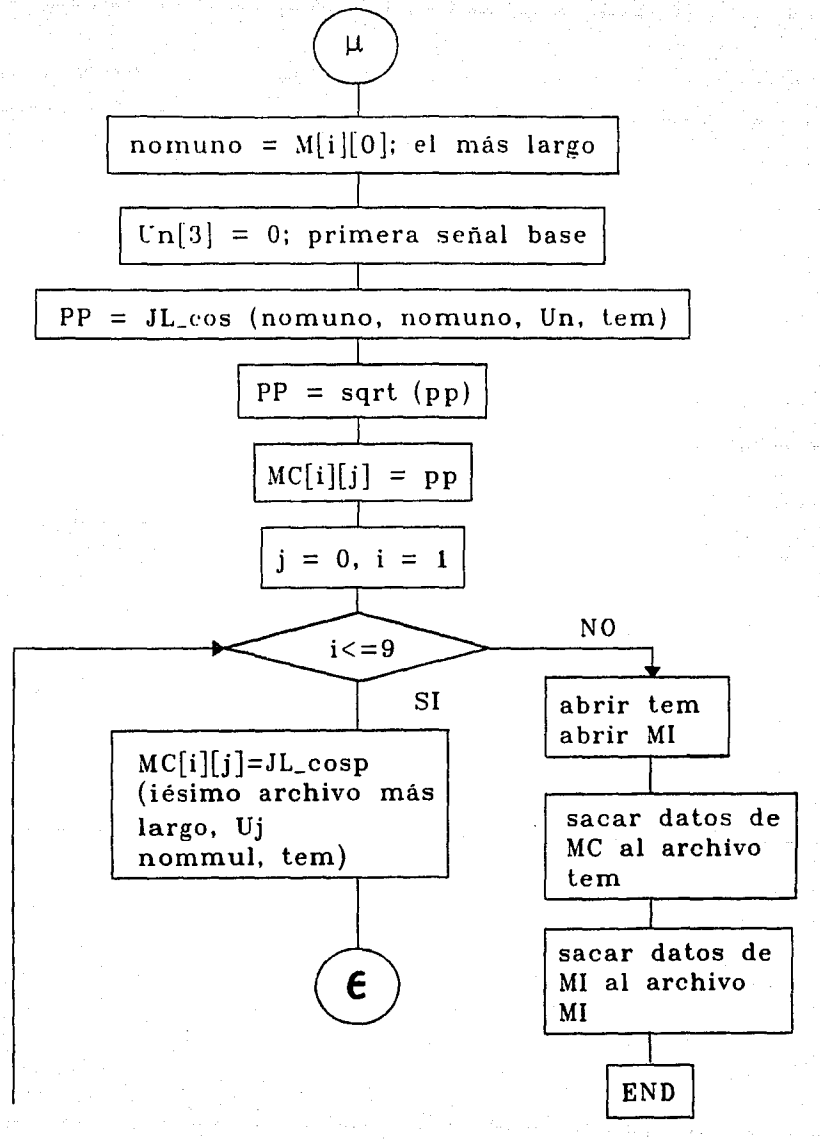


Figura A.5 Diagrama de flujo de JL_subs


nomuno=nombre del primer archivo
a multiplicar
nomdos=nombre del segundo archivo
a multiplicar
Un=nombre del primer archivo unitario
nommul=nombre del archivo multiusos
tem=nombre del archivo con coeficientes
de MC
MI=nombre del archivo con coeficientes
de MI
i = 0,9
NS[i][0]=i
NS[i][1]=número de muestras del
archivo señal i
ns = 1
ß


ß
ren = 0,9
i = 0,9
No
NS[i][1]>=ns
Si
ns = NS[i][1]
ni = i
MI[ren][0] = ni
MI[ren][1] = ns
NS[ni][1] = 0
ns = 1
ren = 0
MC[JC] = 0
i = 0, j = 0
µ


μ
nomuno = M[i][0]; el más largo
Un[3] = 0; primera señal base
PP = JL_cos (nomuno, nomuno, Un, tem)
PP = sqrt (pp)
MC[i][j] = pp
j = 0, i = 1
i<=9
NO
SI
MC[i][j]=JL_cosp (iésimo archivo más largo, Uj nommul, tem)
abrir tem
abrir MI
sacar datos de MC al archivo tem
sacar datos de MI al archivo MI
END
ϵ

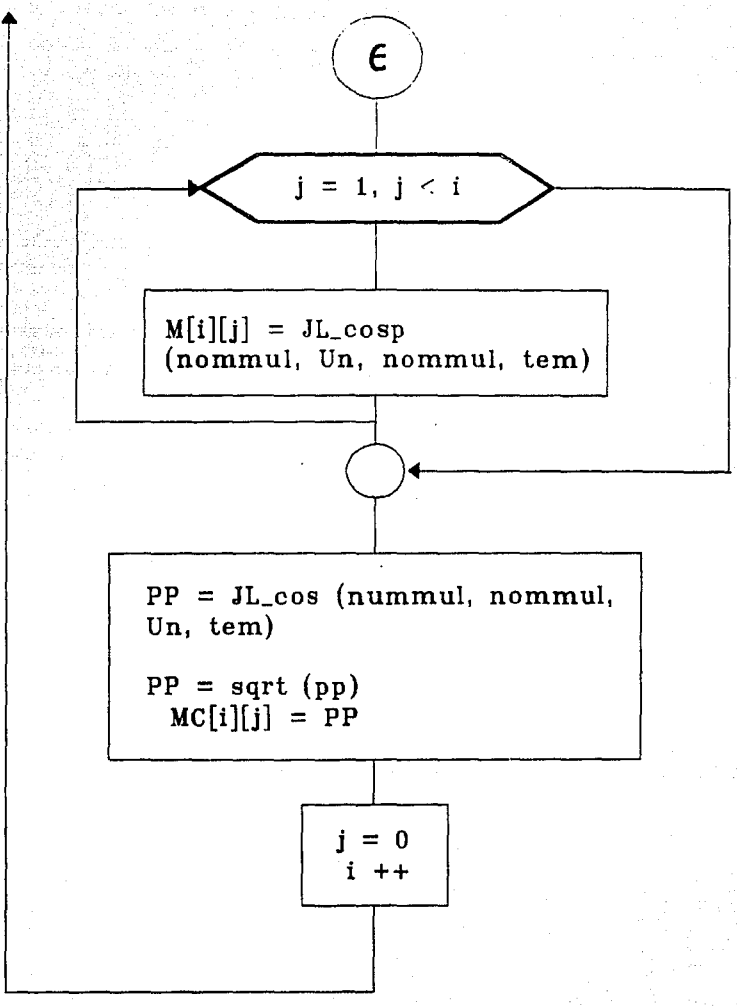
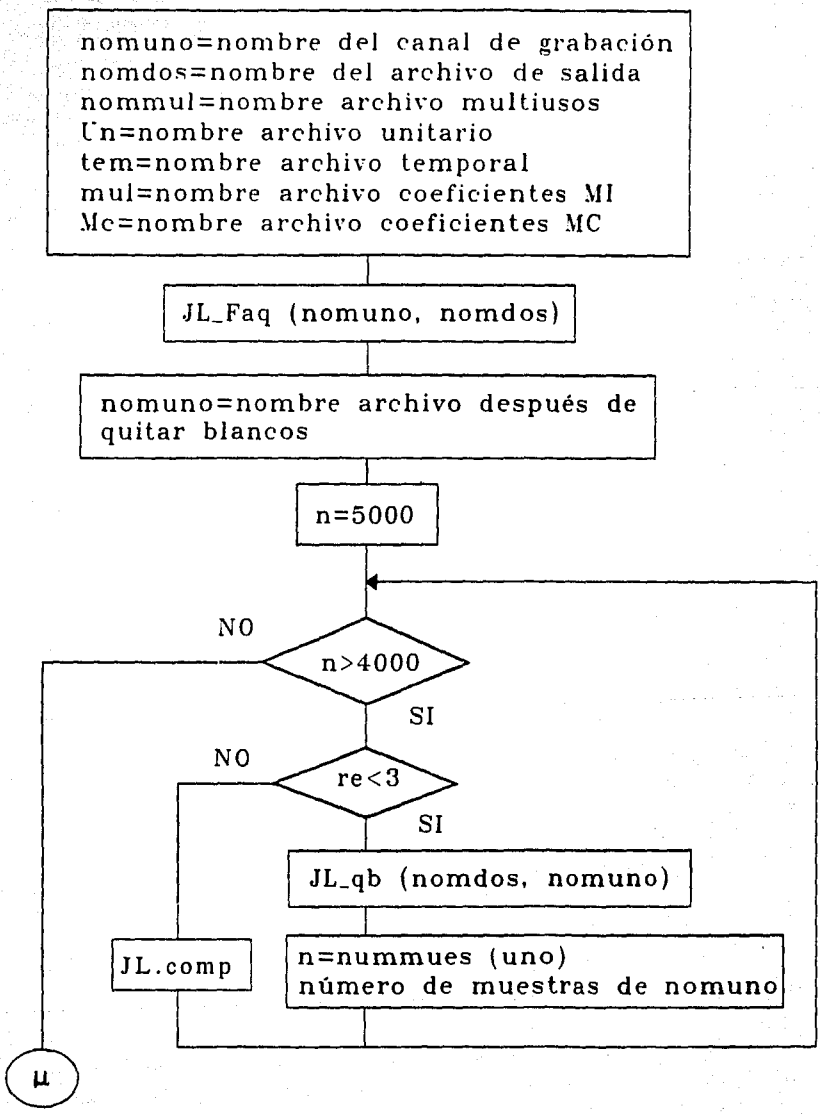


Figura A.6 Procedimiento de Gram-Schmidt

AD.C


nomuno=nombre del canal de grabación
nomdos=nombre del archivo de salida
nommul=nombre archivo multiusos
Un=nombre archivo unitario
tem=nombre archivo temporal
mul=nombre archivo coeficientes MI
Mc=nombre archivo coeficientes MC
JL_Faq (nomuno, nomdos)
nomuno=nombre archivo después de quitar blancos
n=5000
NO
n>4000
SI
NO
re<3
SI
JL_qb (nomdos, nomuno)
JL.comp
n=nummues (uno)
número de muestras de nomuno
μ

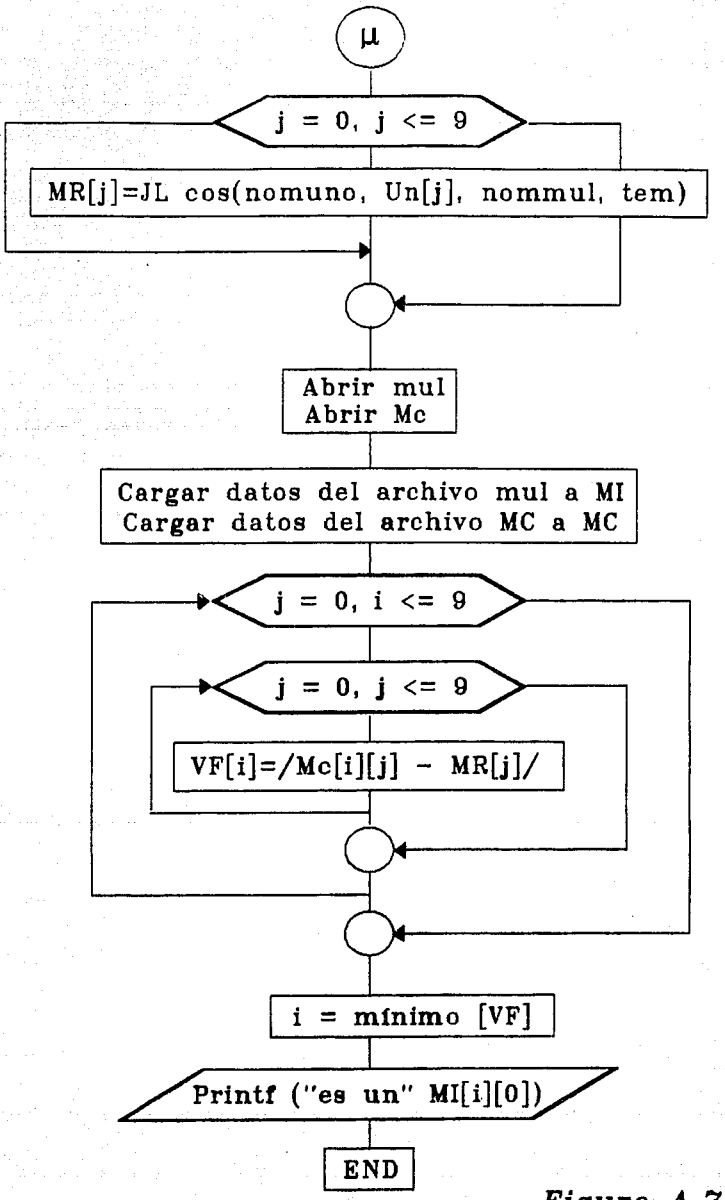


Figura A.7 AD.C

BIBLIOGRAFIA

1. Ahmed N. , Rao K.R.
   Orthogonal Transforms for Digital Signal Processing
   Ed. Springer - Yerlag
   New York - Heidelberg - Berlín
   GDR
   1975

2. Couch II Leon W
   Digital and Analog Communication Systems
   Ed. Macmillan Publishing Company
   Collier Macmillan Publishers
   New York
   USA
   1987

3. Miller Richard K. ,Walker Terri C.
   Natural Language and Voice Processing
   Published by The Fairmont Press, INC.
   USA
   1990

4. Lea Wayne A.
   Trends in Speech Recognition
   Prentice - Hall Inc.
   USA
   1980

5. Witten I. H.
   Principles of Computer Speech
   Academic Press, Inc.
   Harcourt Brace Jovanovich, Publishers
   Orlando, Florida
   USA
   1982

6. Fallside Frank and Woods William A.
   Computer Speech Processing
   Prentice-Hall International
   Great Britain
   1985

7. A tutorial on Speech Understanding Systems
   Invited Papers Presented at the 1974 IEEE Symposium
   Edited by D. Raj Reddy
   Academic Press
   New York-San Francisco-London
   USA
   1975

8. Gibson Jerry D.
Principles of Digital and Analog Communications
Macmillan Publishing Company
USA
1989

9. Tetschner Walt
Voice Processing
Artech House, Inc.
USA
1991

10. Shaughnessy Douglas O.
Speech Communication Human and Machine
Addison-Wesley Publishing Company
USA
1990

11. Rabiner Lawrence R.
A tutorial on Hidden Marrkov Models and
Selected Applications in Speech Recognition
Proceedings of the IEEE
Vol 77 No. 2 February 1989

12. Kun-Shan Lin
Frantz Gene A.
Simar Ray Jr
The TMS320 family of Digital Signal Processors
Proceedings of the IEEE
Vol. 75 No. 9 September 1987

13. Allen Jonathan
A Perspective on Man-Machine
Communication by Speech
Proceedings of the IEEE
Vol. 73 No. 11 November 1985

14. Silverman Hervey F. and Morgan David P.
The Application of Dynamic Programming
to connected Speech Recognition
IEEE ASSP Magazine
Volume 7 November 3 July 1990